

SECURITY IN COMPUTING, FIFTH EDITION

Chapter 5: Operating Systems

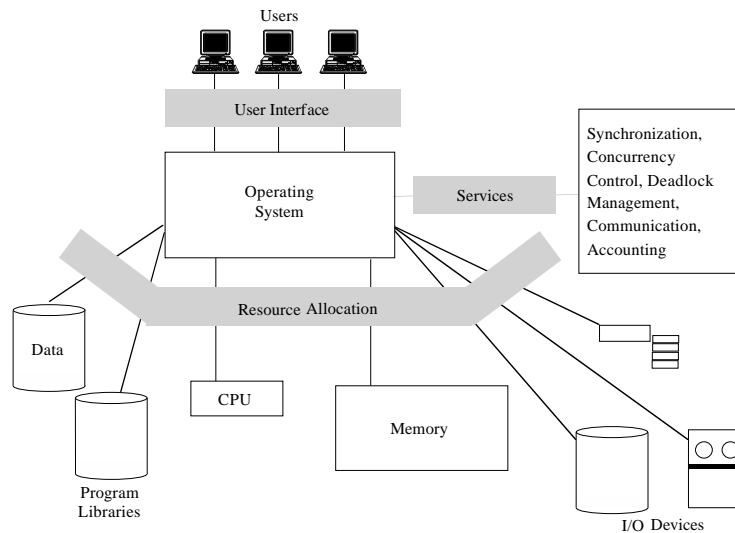
From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Chapter 5 Objectives

- Basic security functions provided by operating systems
- System resources that require operating system protection
- Operating system design principles
- How operating systems control access to resources
- The history of trusted computing
- Characteristics of operating system rootkits

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Operating System Functions



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Security-relevant features:

- Enforced sharing
- Interprocess communication and synchronization
- Protection of critical data
- Guaranteed fair service
- Interface to hardware
- User authentication
- Memory protection
- File and I/O device access control
- Allocation and access control to general objects

History of Operating Systems

- Single-user systems, no OS
- Multiprogrammed OS, aka monitors
 - Multiple users
 - Multiple programs
 - Scheduling, sharing, concurrent use
- Personal computers

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

- First, an entire computer was dedicated to one program at a time, but this approach proved wasteful
- The first operating systems saved startup, loading, and shutdown time and made much better use of limited resources
- The first personal computers took a major step back, as they were dedicated to single users and effectively one program at a time
- Multitasking returned to the mainstream in the 1990s, and with it came all the lessons of the early shared computers

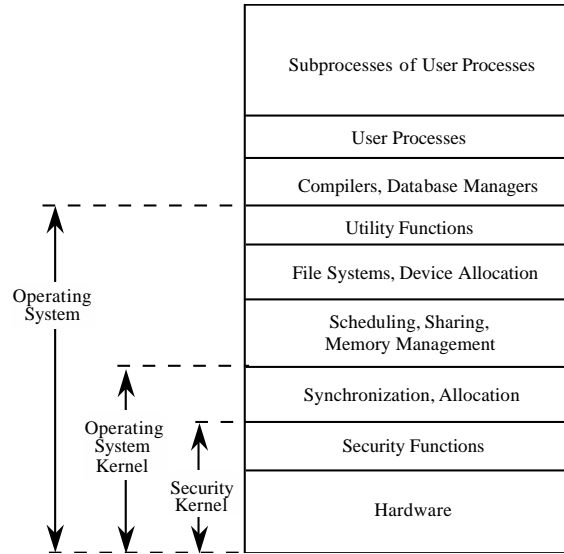
Protected Objects

- Memory
- Sharable I/O devices, such as disks
- Serially reusable I/O devices, such as printers
- Sharable programs and subprocedures
- Networks
- Sharable data

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

These are some of the common objects that need protection by and in OSs.

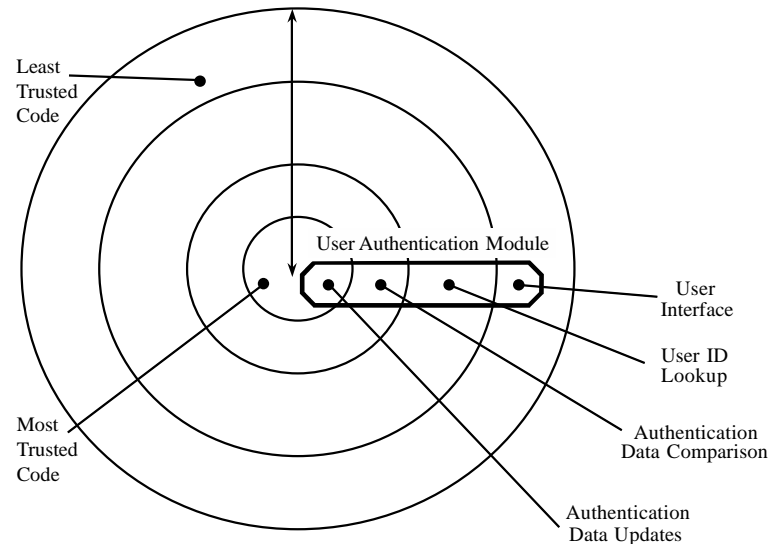
OS Layered Design



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Operating system visualized in layers, from most critical (bottom) to least critical.

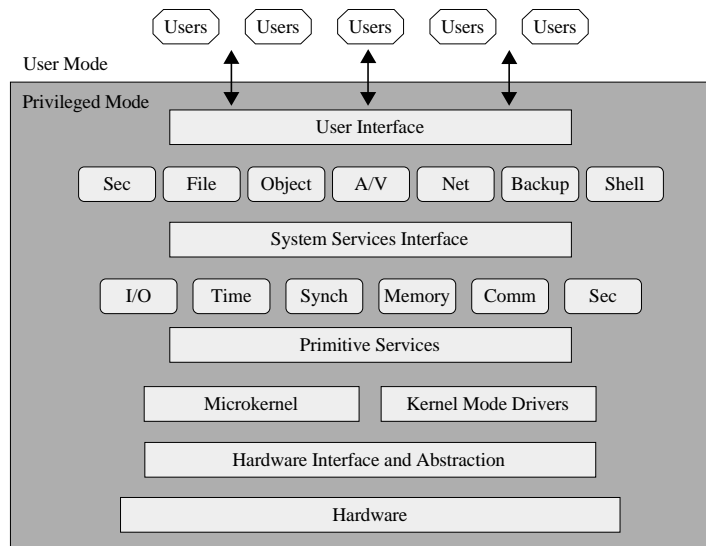
Functions Spanning Layers



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Authentication is a good example of a function that needs to span the layers in the layered model.

Modular OS Design



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Modern OSs are built from discrete modules. These modules generally come from a variety of sources and are subject to updating/overwriting, so they cannot trust one another.

Virtualization

- With virtualization, the OS presents each user with just the resources that user should see
- The user has access to a virtual machine (VM), which contains those resources
- The user cannot access resources that are available to the OS but exist outside the VM
- A hypervisor, or VM monitor, is the software that implements a VM
 - Translates access requests between the VM and the OS
 - Can support multiple OSs in VMs simultaneously
- Honeypot: A VM meant to lure an attacker into an environment that can be both controlled and monitored

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

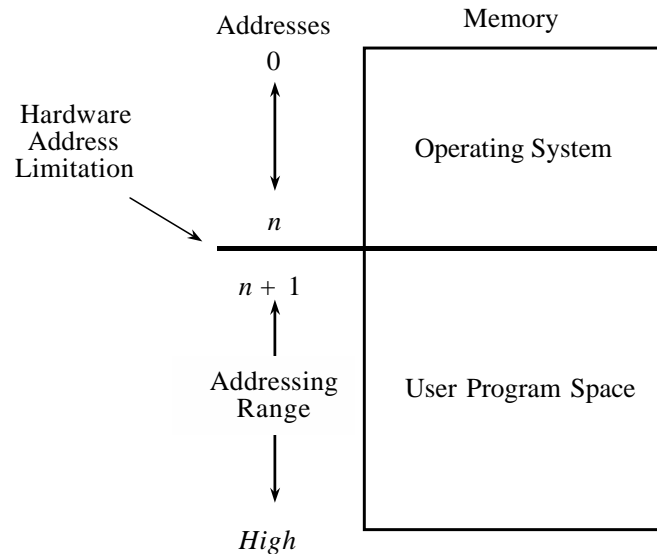
The takeaway here is that, by acting as a sandbox, virtualization is a robust form of access control.

Separation and Sharing

- Methods of separation:
 - Physical
 - Temporal
 - Logical
 - Cryptographic
- Methods of supporting separation/sharing:
 - Do not protect
 - Isolate
 - Share all or share nothing
 - Share but limit access
 - Limit use of an object

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

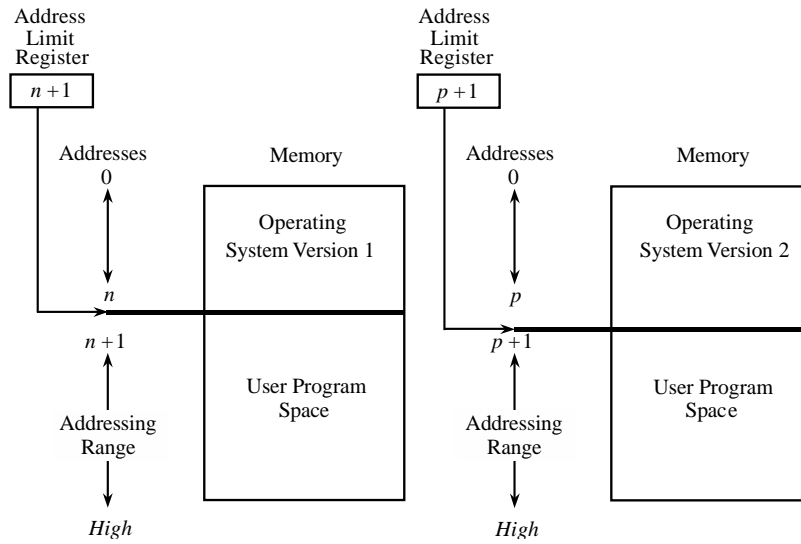
Hardware Protection of Memory



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

A fence defined by a fixed memory address. Users have access only to memory above a certain address.

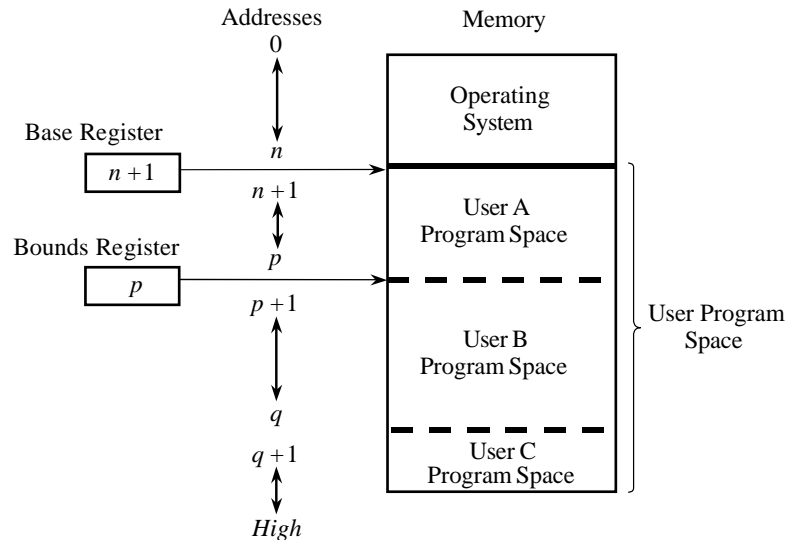
Fence Registers



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Like fences, but fence registers allow for the boundary to change.

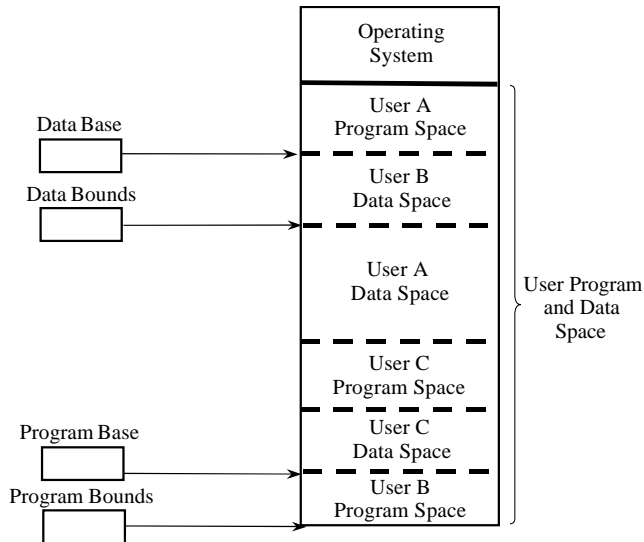
Base/Bounds Registers



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

With base and bounds registers, memory space can be broken into more than two sections, allowing for multiple users.







Two Pairs of Base/Bounds Registers



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

This separates executable memory from data memory for each user, making it harder for bugs/attacks to overwrite code.

Tagged Architecture

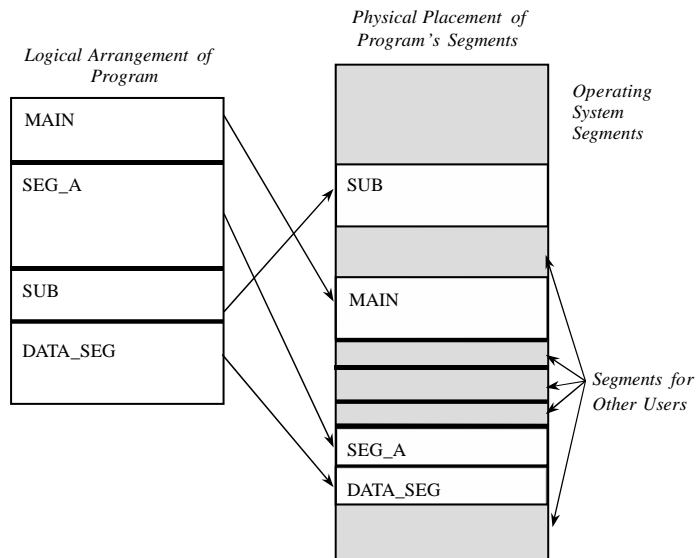
Tag	Memory Word
R	0001
RW	0137
R	0099
X	
X	
X	
X	
X	
X	
R	4091
RW	0002

Code: R = Read-only RW = Read/Write
X = Execute-only

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

In a tagged architecture, each word of machine memory has one or more extra bits to identify its access rights. The big benefit is that access rights aren't based on contiguous memory locations. Tagged architecture has not been widely adopted.

Segmentation

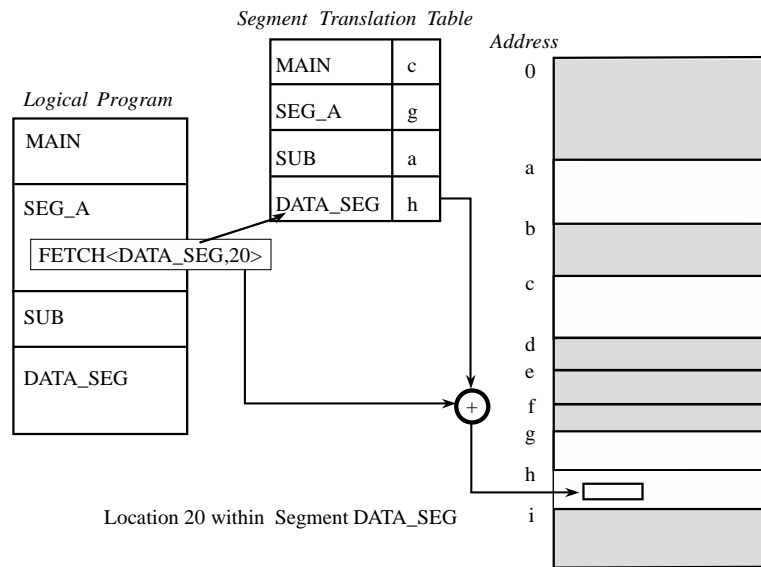


From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

A program is divided into separate, logical pieces (e.g., an array, a procedure). Each segment has its own set of access rights. The operating system maintains a table of each segment and its true memory address, and it translates calls to each segment using that table (shown on next slide). Advantages:

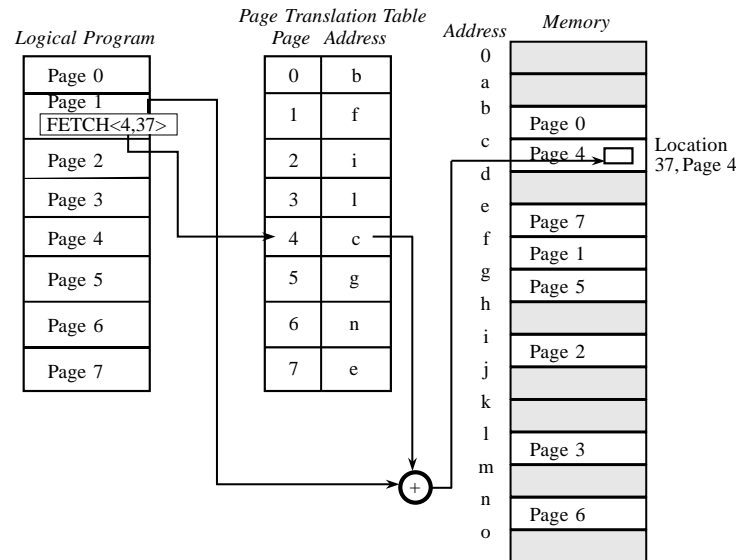
- The operating system can move segments around as necessary, which is very helpful as segments grow and shrink.
- Segments can be removed from memory if they aren't being used currently.
- Every legitimate address reference must pass through the OS, providing an opportunity for access control.

Segment Address Translation



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

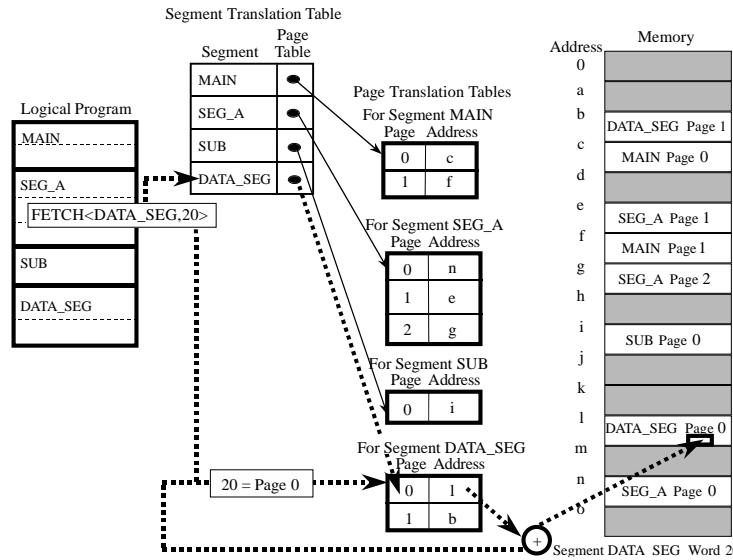
Paging



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Similar to segmentation, but programs are broken into fixed-size fragments (pages) rather than being broken down by logical unit. Because programs aren't broken into logical units, paging doesn't allow different parts of a program to have different access rights.

Paged Segmentation



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Programs can be broken into segments, and the segments are then combined to fill pages. This approach creates an extra layer of translation but allows for the benefits of both paging and segmentation.

Principles of Secure OS Design

- Simplicity of design
 - OSs are inherently complex, and any unnecessary complexity only makes them harder to understand and secure
- Layered design
 - Enables layered trust
- Layered trust
 - Layering is both a way to keep a design logical and understandable and a way to limit risk
 - Example: very tight access controls on critical OS functions, fewer access controls on important noncritical functions, and few if any access controls on functions that aren't important to the OS

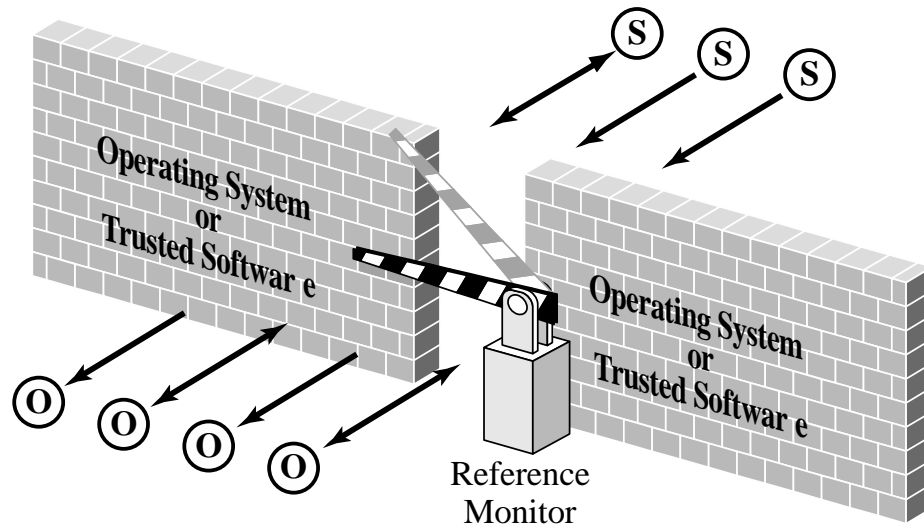
From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Kernelized Design

- A kernel is the part of the OS that performs the lowest-level functions
 - Synchronization
 - Interprocess communication
 - Message passing
 - Interrupt handling
- A security kernel is responsible for enforcing the security mechanisms of the entire OS
 - Typically contained within the kernel

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Reference Monitor



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

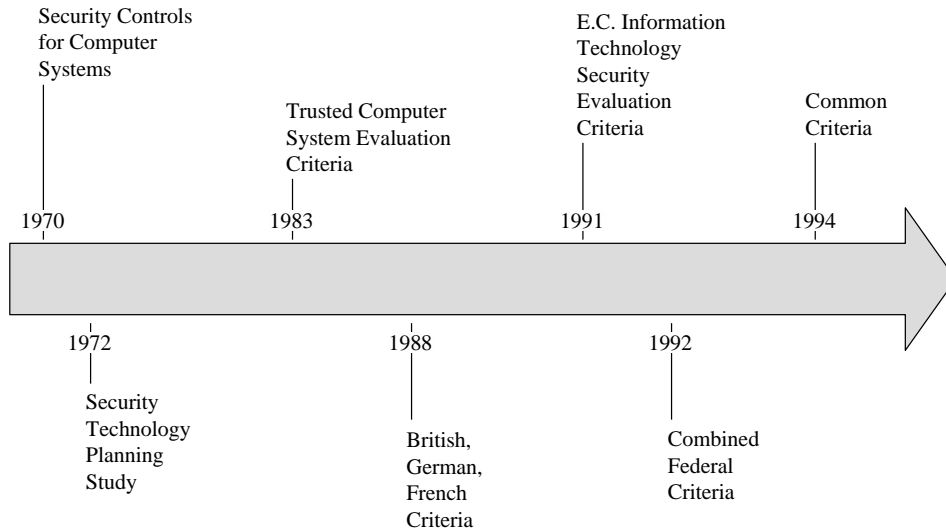
The reference monitor is the most important part of the security kernel, controlling access to objects. A reference monitor must be tamperproof, unbypassable, and analyzable.

Trusted Systems

- A trusted system is one that has been shown to warrant some degree of trust that it will perform certain activities faithfully
- Characteristics of a trusted system:
 - A defined policy that details what security qualities it enforces
 - Appropriate measures and mechanisms by which it can enforce security adequately
 - Independent scrutiny or evaluation to ensure that the mechanisms have been selected and implemented properly

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

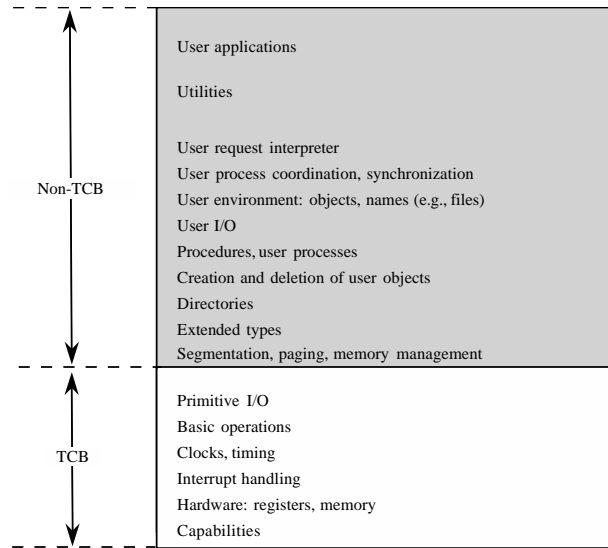
History of Trusted Systems



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Attempts to declare computers trustworthy go back almost 50 years. Over the years, changes in technology have resulted in new requirements, and the explosion of new devices and software have made it impossible to keep up.

Trusted Computing Base (TCB)



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

The TCB portion of the OS is the part we depend on for enforcement of security policy. The TCB monitors and protects the secrecy and integrity of four basic interactions: process activation, execution domain switching, memory protection, and I/O operation.

Other Trusted System Characteristics

- Secure startup
 - System startup is a tricky time for security, as most systems load basic I/O functionality before being able to load security functions
- Trusted path
 - An unforgeable connection by which the user can be confident of communicating directly with the OS
- Object reuse control
 - OS clears memory before reassigning it to ensure that leftover data doesn't become compromised
- Audit
 - Trusted systems track security-relevant changes, such as installation of new programs or OS modification
 - Audit logs must be protected against tampering and deletion

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

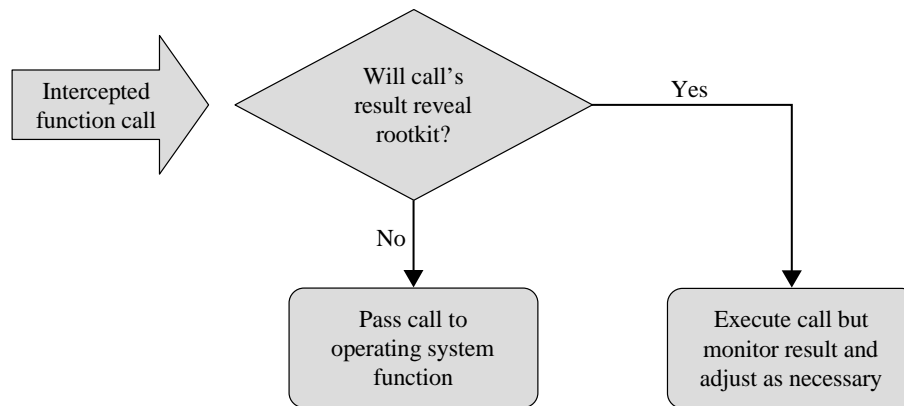
Intel's trusted boot technology uses TPMs to achieve secure startup.

Rootkits

- A rootkit is a malicious software package that attains and takes advantage of root status or effectively becomes part of the OS
- Rootkits often go to great length to avoid being discovered or, if discovered and partially removed, to reestablish themselves
 - This can include intercepting or modifying basic OS functions

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Rootkit Evading Detection



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

In this example, a rootkit is hooking a system call in order to intercept potentially threatening results.

Summary

- OSs have evolved from supporting single users and single programs to many users and programs at once
- Resources that require OS protection: memory, I/O devices, programs, and networks
- OSs use layered and modular designs for simplification and to separate critical functions from noncritical ones
- Resource access control can be enforced in a number of ways, including virtualization, segmentation, hardware memory protection, and reference monitors
- Rootkits are malicious software packages that attain root status or effectively become part of the OS

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.