NC STATE UNIVERSITY Computer Science

CSC 405
Introduction to Computer Security

Topic 5. Trusted Operating Systems
-- Part II

Trusted vs. Trustworthy

- A component of a system is trusted means that
  - the security of the system depends on it
  - if the component is insecure, so is the system
  - determined by its role in the system
- A component is trustworthy means that
  - the component deserves to be trusted
    - e.g., it is implemented correctly
  - determined by intrinsic properties of the component

Trusted Operating System is actually a misnomer

Terminology: Trusted Computing Base

- The set of all hardware, software and procedural components that enforce the security policy
  - in order to break security, an attacker must subvert one or more of them
- What consists of the conceptual Trusted Computing Based in a Unix/Linux system?
  - Hardware, kernel, system binaries, system configuration files, etc.

Terminology: Trusted Path

- Mechanism that provides confidence that the user is communicating with what the user intended to communicate with (typically TCB)
  - attackers can't intercept or modify whatever information is being communicated
  - defends attacks such as fake login programs
- Example: Ctrl+Alt+Del for log in on Windows

Terminology: Trusted Computing

- Trusted Computing Group
  - an alliance of Microsoft, Intel, IBM, HP and AMD which promotes a standard for a `more secure' PC.
  - formally TCPA
- Next-Generation Secure Computing Base (NGSCB) by Microsoft
  - formally Palladium
  - intend to provide strong process isolation, sealed storage, secure path to and from the user, and attestation
- Ensure that users can't tamper with the application software, and these applications can communicate securely with their authors and with each other
  - driven by Digital Right Management needs

What Makes a "Trusted" OS

- Extra security features (compared to ordinary OS)
  - Stronger authentication mechanisms
    - Example: require token + password
  - More security policy options
    - Example: only let users read file f for purpose p
  - Logging and other features
- More secure implementation
  - Apply secure design and coding principles
  - Assurance and certification
    - Code audit or formal verification
  - Maintenance procedures
    - Apply patches, etc.

Trusted OS Design

- Security must be considered in every aspect of its design
  - There must be a clear mapping from security requirements to the design
  - Validate that the design has been done correctly
    - A mapping from security requirements to design to tests
- Security must be an essential part of the initial design
  - Security considerations may shape many of the OS design decisions

Good Design Principles

- Least privilege
  - Each user and each program should operate using the fewest privileges possible to complete the job
- Economy of mechanism
  - Keep the design small, simple, and straightforward
- Open design
  - Security does not depend on the secrecy of mechanism
- Complete mediation
  - every access must be checked

Good Design Principles (Cont'd)

- Permission based
  - The default condition should be denial of access
- Separation of privilege
  - Access to objects should depend on more than on condition (e.g., user authentication + crypto)
  - Someone who defeats one protection system will not have full access
- Least common mechanism
  - Minimize the amount of mechanism common to more than one user and depended on by all users
  - Physical and logical separation⇒reduce the risk from sharing
- Ease of use
  - Human interface should be designed for ease of use
  - If a protection mechanism is easy to use, it's unlikely to be avoided

Security Features of An OS

- Security features of an ordinary OS
- Security features of a trusted OS

Security Features of An Ordinary OS

- User authentication
- Protection of memory
- File and I/O device access control
- Allocation and access control to general objects
- Enforcement of sharing
- Guarantee of fair service
- Inter-process communication and synchronization
- Protection of OS protection data

An ordinary OS provides a set of security features

Security Features of A Trusted OS

- Identification and authentication
- Mandatory access control
  - MAC not under user control, precedence over DAC
- Object reuse protection
  - Write over old data when file space is allocated
- Complete mediation
  - Prevent any access that circumvents monitor
- Trusted path
  - An unmistakable communication channel with the OS

A trusted OS provides a set of security features together with an appropriate degree of assurance that the features have been assembled and implemented correctly.

Security Features of A Trusted OS (Cont'd)

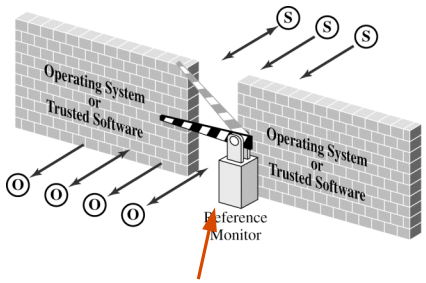
- Accountability and Audit
  - Log security-related events and check logs
- Audit log reduction
  - Reduce the volume of audit data
  - Find useful information
- Intrusion detection
  - Anomaly detection: Learn normal activity, and report abnormal
  - Misuse detection: Recognize patterns of known attacks

Kernelized Design

- Design of a security kernel for trusted OS
  - A security kernel is responsible for enforcing the security mechanisms of the entire OS
  - The security kernel is contained in the OS kernel
- Two design choices
  - Security kernel is isolated and used as an addition to the OS
  - Security kernel forms the basis of the entire OS

Reference Monitor

- The collection of access controls for devices, files, memory, IPC, and other objects
- Must be
  - Tamperproof
  - Cannot be bypassed
  - Small enough for analysis
  - Complete
- Work along with other security mechanisms
  - E.g., audit



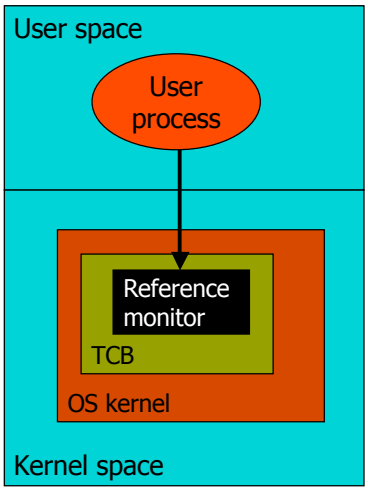
Reference monitor

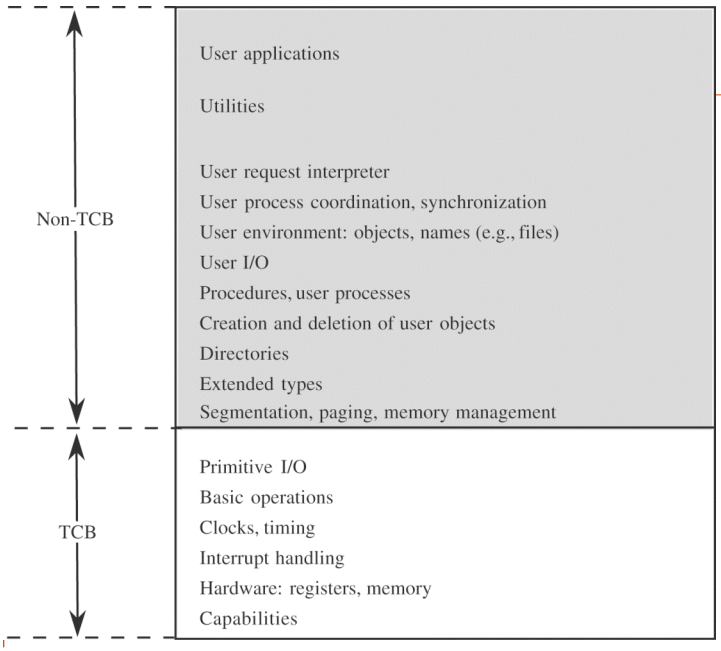
Trusted Computing Base (TCB)

- TCB consists of the parts of the trusted OS on which we depend for correct enforcement of policy
- TCB components
  - Hardware -- processor, memory, registers, I/O devices
  - Some notion of processes -- allows separation
  - Primitive files -- access control files, id/authentication data
  - Protected memory -- reference monitor can be protected
  - IPC -- different parts of TCB can pass data to activate other parts

Reference Monitor and TCB

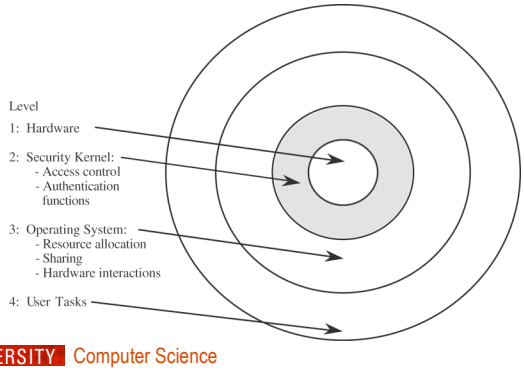
- Reference monitor
  - May be a part of TCB
  - All system calls go through reference monitor for security checking
  - Most OS not designed this way





TCB Design and Implementation

- One sensible approach
  - Design the TCB first, and design the OS around it
  - Example: Honeywell's secure OS Scomp
    - TCB: 20 modules; <1,000 lines of source code
    - Whole system: about 10,000 lines of code



Separation/Isolation

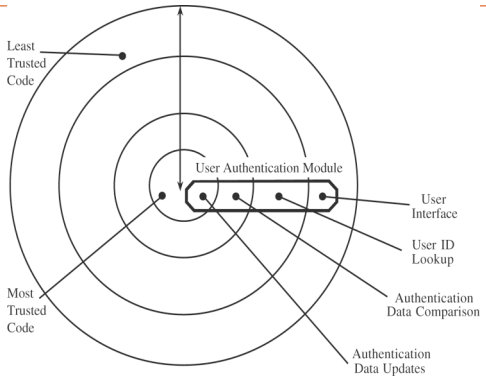
- Physical separation
  - Different hardware
- Temporal separation
  - Different times
- Cryptographic separation
  - Cryptographic protection of sensitive data
- Logical separation
  - Isolation
  - Reference monitor separates one user's objects from another user

Virtualization

- OS simulates or emulates a collection of computer system's resources
  - Virtual machine
    - Processor
    - Instruction set
    - Storage
    - I/O devices
  - Gives the user a full set of hardware features
  - Note the difference from virtual memory
  - Example: IBM Processor Resources/System Manager (PR/SM) system

Layered Design

- Motivation
  - Implementing all operations of a function in security kernel
    - Not all operations warrant high security
- Layered design
  - Implement a single logical function in several different modules in different layers



Closer to the center -- more trusted and more sensitive

Inner layers present functionality to outer layers;
Each layer encapsulates those inside it.

Assurance Methods

- Testing and penetration testing
  - Can demonstrate existence of flaws, not absence
- Formal verification
  - Time-consuming, painstaking process
- "Validation"
  - Requirements checking
    - Demonstrate that the system does each thing listed in the requirements
  - Design and code reviews
    - Scrutinize the design or the code (each requirement --> design and code); note problems along the way
  - Module and system testing
    - Acceptance testing: Confirm each requirement in the system

Assurance Criteria

- Criteria are specified to enable evaluation
- Originally motivated by military applications, but now is much wider
- Examples
  - Orange Book
  - Common Criteria

TCSEC: 1983–1999

- Trusted Computer System Evaluation Criteria
  - Also known as the Orange Book
  - Series that expanded on Orange Book in specific areas was called *Rainbow Series*
  - Developed by National Computer Security Center, US Dept. of Defense
- Heavily influenced by Bell-LaPadula model and reference monitor concept
- Emphasizes confidentiality

Evaluation Classes C and D

D Did not meet requirements of any other class

C1 *Discretionary protection*; minimal functional, assurance requirements; I&A controls; DAC

C2 *Controlled access protection*; object reuse, auditing, more stringent security testing

B1 *Labeled security protection*; informal security policy model; MAC for some objects; labeling; more stringent security testing

Evaluation Classes A and B

B2 *Structured protection*; formal security policy model; MAC for all objects, labeling; trusted path; least privilege; covert channel analysis, configuration management

B3 *Security domains*; full reference validation mechanism; increases trusted path requirements, constrains code development; more DTLS requirements; documentation

A1 *Verified protection*; significant use of formal methods; trusted distribution; code, FTLS correspondence

Limitations

- Written for operating systems
  - NCSC introduced "interpretations" for other things such as networks (*Trusted Network Interpretation*, the Red Book), databases (*Trusted Database Interpretation*, the Purple or Lavender Book)
- Focuses on BLP
  - Most commercial firms do not need MAC
- Does not address integrity or availability
  - Critical to commercial firms
- Combine functionality and assurance in a single linear scale

Contributions

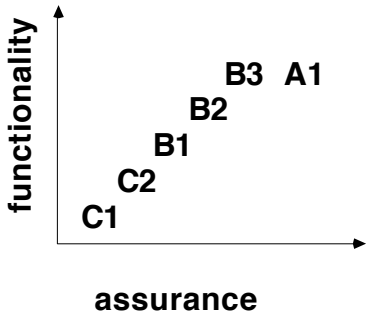
- Heightened awareness in commercial sector to computer security needs
- Commercial firms could not use it for their products
  - Did not cover networks, applications
  - Led to wave of new approaches to evaluation
  - Some commercial firms began offering certifications
- Basis for several other schemes, such as Federal Criteria, Common Criteria

Orange Book Classes -- Unofficial View

C1,C2	Simple enhancement of existing systems. No breakage of applications
B1	Relatively simple enhancement of existing systems. Will break some applications.
B2	Relatively major enhancement of existing systems. Will break many applications.
B3	Failed A1
A1	Top down design and implementation of a new system from scratch

Functionality V.S. Assurance

- **functionality is multi-dimensional**
- **assurance has a linear progression**



Common Criteria: 1998–Present

- Began in 1998 with signing of Common Criteria Recognition Agreement with 5 signers
  - US, UK, Canada, France, Germany
- As of May 2002, 10 more signers
  - Australia, Finland, Greece, Israel, Italy, Netherlands, New Zealand, Norway, Spain, Sweden; India, Japan, Russia, South Korea developing appropriate schemes
- Standard 15408 of International Standards Organization
- *De facto* US security evaluation standard, replaces TCSEC

Common Criteria

- Three parts
  - CC Documents
    - Protection profiles: requirements for category of systems
      - Functional requirements
      - Assurance requirements
  - CC Evaluation Methodology
  - National Schemes (local ways of doing evaluation)

http://www.commoncriteria.org/

CC Functional Requirements

- Contains 11 classes of functional requirements
  - Each contain one or more families
  - Elaborate naming and numbering scheme
- Classes
  - Security Audit, Communication, Cryptographic Support, User Data Protection, Identification and Authentication, Security Management, Privacy, Protection of Security Functions, Resource Utilization, TOE Access, Trusted Path
- Families of Identification and Authentication
  - Authentication Failures, User Attribute Definition, Specification of Secrets, User Authentication, User Identification, and User/Subject Binding

CC Assurance Requirements

- Ten security assurance classes
- Classes:
  - Protection Profile Evaluation
  - Security Target Evaluation
  - Configuration Management
  - Delivery and Operation
  - Development
  - Guidance Documentation
  - Life Cycle
  - Tests
  - Vulnerabilities Assessment
  - Maintenance of Assurance

Protection Profiles (PP)

- "A CC protection profile (PP) is an implementation-independent set of security requirements for a category of products or systems that meet specific consumer needs"
  - Subject to review and certified
- Requirements
  - Functional
  - Assurance
  - EAL

Protection Profiles

- Example: Controlled Access PP (CAPP_V1.d)
  - Security functional requirements
    - Authentication, User Data Protection, Prevent Audit Loss
  - Security assurance requirements
    - Security testing, Admin guidance, Life-cycle support, …
  - Assumes non-hostile and well-managed users
  - Does not consider malicious system developers

Security Targets (ST)

- "A security target (ST) is a set of security requirements and specifications to be used for evaluation of an identified product or system"
- Can be based on a PP or directly from CC
- Describes specific security functions and mechanisms

Evaluation Assurance Levels 1 – 4

EAL 1: Functionally Tested
- Review of functional and interface specifications
- Some independent testing

EAL 2: Structurally Tested
- Analysis of security functions, incl high-level design
- Independent testing, review of developer testing

EAL 3: Methodically Tested and Checked
- Development environment controls; config mgmt

EAL 4: Methodically Designed, Tested, Reviewed
- Informal spec of security policy, Independent testing

Evaluation Assurance Levels 5 – 7

EAL 5: Semiformally Designed and Tested
- Formal model, modular design
- Vulnerability search, covert channel analysis

EAL 6: Semiformally Verified Design and Tested
- Structured development process

EAL 7: Formally Verified Design and Tested
- Formal presentation of functional specification
- Product or system design must be simple
- Independent confirmation of developer tests

Example: Windows 2000, EAL 4+

- Evaluation performed by SAIC
- Used "Controlled Access Protection Profile"
- Level EAL 4 + Flaw Remediation
  - "EAL 4 … represents the highest level at which products not built specifically to meet the requirements of EAL 5-7 …"
    (EAL 5-7 requires more stringent design and development procedures …)
  - Flaw Remediation
- Evaluation based on specific configurations
  - Produced configuration guide that may be useful

Is Windows "Secure"?

- Good things
  - Design goals include security goals
  - Independent review, configuration guidelines
- But …
  - "Secure" is a complex concept
    - What properties protected against what attacks?
  - Typical installation includes more than just OS
    - Many problems arise from applications, device drivers
  - Security depends on installation as well as system