

TEL2813/IS2820

Security Management

Systems/Evaluations
Lecture 11
April 7, 2005

Access control matrix

Objects

	File 1	File 2	File 3	...	File n
User 1	read	write	-	-	read
User 2	write	write	write	-	-
User 3	-	-	-	read	read
...		write	write	write	
User m	read	write	read	write	read

Subjects



Two implementation concepts

- Access control list (ACL)
 - Store column of matrix with the resource
- Capability
 - User holds a “ticket” for each resource
 - Two variations
 - store row of matrix with user
 - unforgeable ticket in user space



Unix

- Developed at AT&T Bell Labs
- Single monolithic kernel
 - Kernel mode
 - File system, device drivers, process management
 - User programs run in user mode
 - networking



Unix Identification and authentication

- Users have username
 - Internally identified with a user ID (UID)
 - Username to UID info in /etc/passwd
 - Super UID = 0
 - can access any file
 - Every user belong to a group – has GID
- Passwords to authenticate
 - in /etc/passwd
 - Shadow file /etc/shadow



Unix file security

- Each file has owner and group
- Permissions set by owner
 - Read, write, execute
 - Owner, group, other
 - Represented by vector of four octal values
- Only owner, root can change permissions
 - This privilege cannot be delegated or shared

Unix File Permissions

- File type, owner, group, others

```
drwx----- 2 jjoshi isfac 512 Aug 20 2003 risk management
lrwxrwxrwx 1 jjoshi isfac 15 Apr 7 09:11 risk_m->risk management
-rw-r--r-- 1 jjoshi isfac 1754 Mar 8 18:11 words05.ps
-r-sr-xr-x 1 root bin 9176 Apr 6 2002 /usr/bin/rs
-r-sr-sr-x 1 root sys 2196 Apr 6 2002 /usr/bin/passwd
```

- File type: regular -, directory d, symlink l, device b/c, socket s, fifo f/p
- Permission: r, w, x, s or S (set.id), t (sticky)
- While accessing files
 - Process EUID compared against the file UID
 - GIDs are compared; then Others are tested


Effective user id (EUID)

- Each process has three Ids
 - Real user ID (RUID)
 - same as the user ID of parent (unless changed)
 - used to determine which user started the process
 - Effective user ID (EUID)
 - from set user ID bit on the file being executed, or sys call
 - determines the permissions for process
 - Saved user ID (SUID)
 - Allows restoring previous EUID
- Similarly we have
 - Real group ID, effective group ID,



IDs/Operations

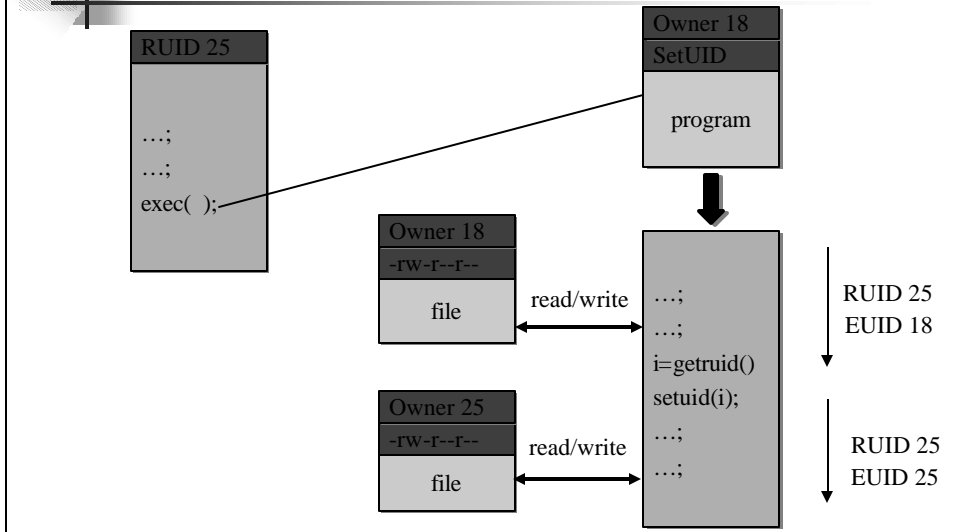
- Root can access any file
- Fork and Exec
 - Inherit three IDs,
 - except exec of file with setuid bit
- Setuid system calls
 - seteuid(newid) can set EUID to
 - Real ID or saved ID, regardless of current EUID
 - Any ID, if EUID=0
 - Related calls: setuid, seteuid, setreuid



Setid bits on executable Unix file

- Three setid bits
 - Setuid
 - set EUID of process to ID of file owner
 - Setgid
 - set EGID of process to GID of file
 - Setuid/Setgid used when a process executes a file
 - If setuid (setgid) bit is on – change the EUID of the process changed to UID (GUID) of the file
 - Sticky
 - Off: if user has write permission on directory, can rename or remove files, even if not owner
 - On: only file owner, directory owner, and root can rename or remove file in the directory

Example



Careful with Setuid !

- Can do anything that owner of file is allowed to do
- Be sure not to
 - Take action for untrusted user
 - Return secret data to untrusted user
- Principle of least privilege
 - change EUID when root privileges no longer needed
- Setuid scripts (bad idea)
 - Race conditions: begin executing setuid program; change contents of program before it loads and is executed

Anything possible if root



Windows NT

- Windows 9x, Me
 - Never meant for security
 - FAT file system – no file level security
 - PWL password scheme – not secure
 - Can be simply deleted
- Windows NT
 - Username mapped to Security ID (SID)
 - SID is unique within a domain
 - SID + password stored in a database handled by the Security Accounts Manager (SAM) subsystem

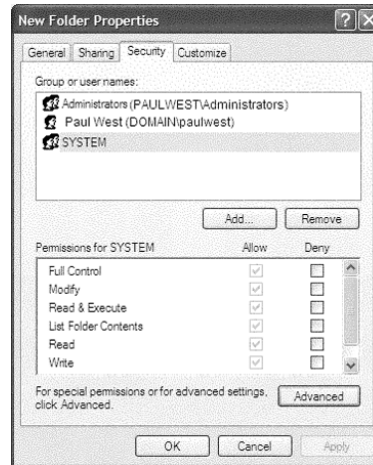


Windows NT

- Some basic functionality similar to Unix
 - Specify access for groups and users
 - Read, modify, change owner, delete
- Some additional concepts
 - Tokens
 - Security attributes
- Generally
 - More flexibility than Unix
 - Can define new permissions
 - Can give some but not all administrator privileges

Sample permission options

- SID
 - Identity (replaces UID)
 - SID revision number
 - 48-bit authority value
 - variable number of Relative Identifiers (RIDs), for uniqueness
 - Users, groups, computers, domains, domain members all have SIDs



Permission Inheritance

- Static permission inheritance (Win NT)
 - Initially, subfolders inherit permissions of folder
 - Folder, subfolder changed independently
 - *Replace Permissions on Subdirectories* command
 - Eliminates any differences in permissions



Permission Inheritance

- Dynamic permission inheritance (Win 2000)
 - Child inherits parent permission, remains linked
 - Parent changes are inherited, except explicit settings
 - Inherited and explicitly-set permissions may conflict
 - Resolution rules
 - Positive permissions are additive
 - Negative permission (deny access) takes priority



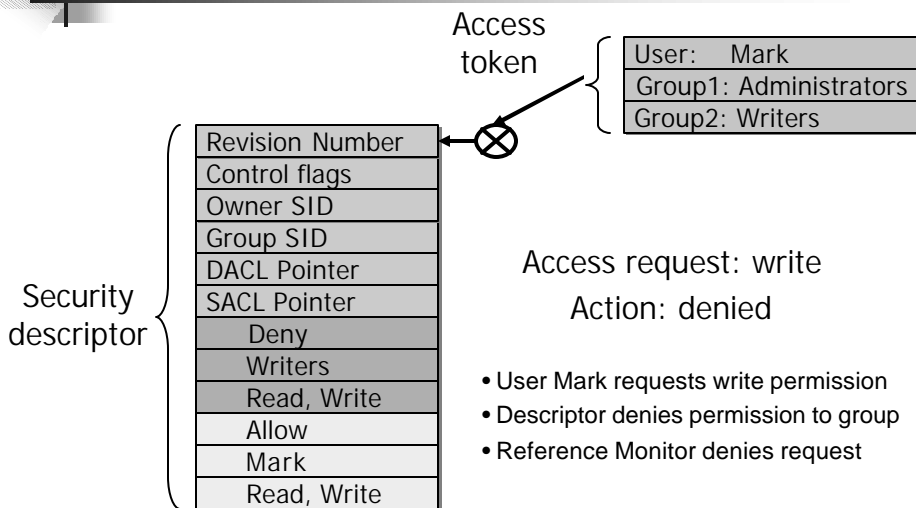
Tokens

- Security context
 - privileges, accounts, and groups associated with the process or thread
- Security Reference Monitor
 - uses tokens to identify the security context of a process or thread
- Impersonation token
 - Each thread can have two tokens – primary & impersonation
 - thread uses temporarily to adopt a different security context, usually of another user

Security Descriptor

- Information associated with an object
 - who can perform what actions on the object
- Several fields
 - Header
 - Descriptor revision number
 - Control flags, attributes of the descriptor
 - E.g., memory layout of the descriptor
 - SID of the object's owner
 - SID of the primary group of the object
 - Two attached optional lists:
 - Discretionary Access Control List (DACL) – users, groups, ...
 - System Access Control List (SACL) – system logs, ..

Example access request



Impersonation Tokens (setuid?)

- Process uses security attributes of another
 - Client passes impersonation token to server
- Client specifies impersonation level of server
 - Anonymous
 - Token has no information about the client
 - Identification
 - server obtains the SIDs of client and client's privileges, but server cannot impersonate the client
 - Impersonation
 - server identifies and impersonate the client
 - Delegation
 - lets server impersonate client on local, remote systems

Encrypted File Systems (EFS)

- Store files in encrypted form
 - Key management: user's key decrypts file
 - Useful protection if someone steals disk
- Windows – EFS
 - User marks a file for encryption
 - Unique file encryption key is created
 - Key is encrypted, can be stored on smart card

SELinux Security Policy

Abstractions

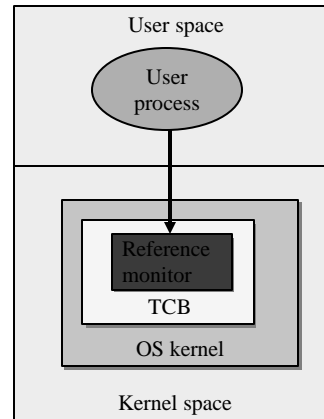
- Type enforcement
 - Each process has an associated domain
 - Each object has an associated type
 - Configuration files specify
 - How domains are allowed to access types
 - Allowable interactions and transitions between domains
- Role-based access control
 - Each process has an associated role
 - Separate system and user processes
 - configuration files specify
 - Set of domains that may be entered by each role

Sample Features of Trusted OS

- 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
- Audit
 - Log security-related events
- Intrusion detection
 - Anomaly detection
 - Learn normal activity, Report abnormal actions
 - Attack detection
 - Recognize patterns associated with known attacks

Kernelized Design

- Trusted Computing Base
 - Hardware and software for enforcing security rules
- Reference monitor
 - Part of TCB
 - All system calls go through reference monitor for security checking
 - Most OS not designed this way
- Reference validation mechanism –
 1. Tamperproof
 2. Never be bypassed
 3. Small enough to be subject to analysis and testing – the completeness can be assured



Is Windows is "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
 - Windows driver certification program



Window 2000

- Newer features than NT
- NTFS file system redesigned for performance
- Active directory
 - Kerberos for authentication
 - IPSec/L2TP



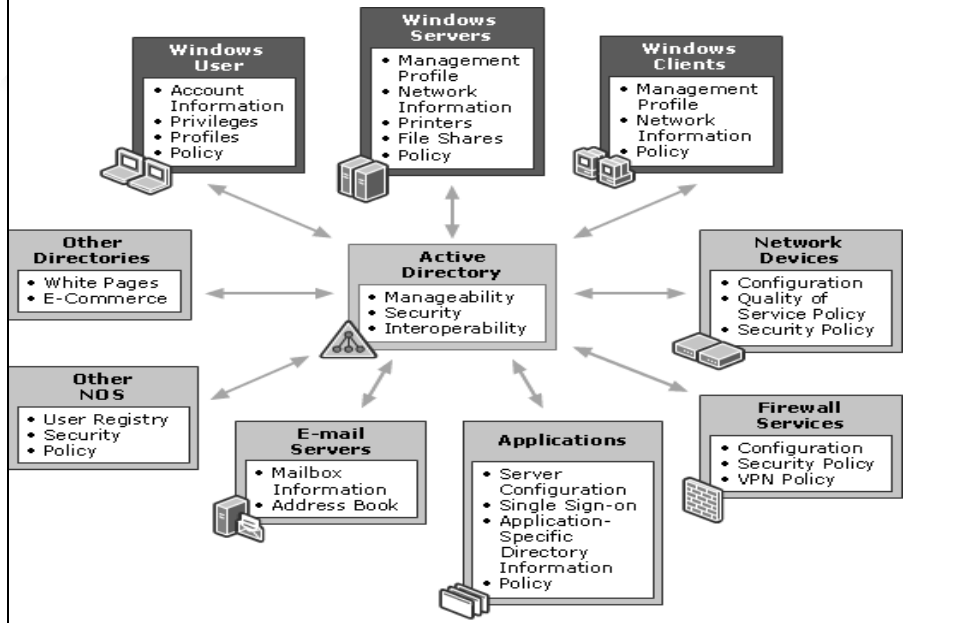
Windows XP

- Improvement over Win 2000 Professional
 - Personalized login
 - Multiple users to have secure profiles
 - User switching
 - Multiple users to be logged in
 - Internet connection firewall (ICF)
 - Active packet filtering
 - Blank password restriction (null sessions)
 - Encrypting File System (EFS) using PKI
 - Smart card support (uses X.509 certificate for authentication)

Active Directory

- Core for the flexibility of Win2000
 - Centralized management for clients, servers and user accounts
- Information about all objects
- Group policy and remote OS operations
- Replaces SAM database
 - AD is trusted component of the LSA
- Stores
 - Access control information – authorization
 - User credentials – authentication
- Supports
 - PKI, Kerberos and LDAP

Win 2003





Evaluation




What is Formal Evaluation?

- Method to achieve *Trust*
 - Not a guarantee of security
- Evaluation methodology includes:
 - Security requirements
 - Assurance requirements showing how to establish security requirements met
 - Procedures to demonstrate system meets requirements
 - Metrics for results (level of trust)
- Examples: TCSEC (Orange Book), ITSEC, CC



Formal Evaluation: Why?

- Organizations require assurance
 - Defense
 - Telephone / Utilities
 - "Mission Critical" systems
- Formal verification of entire systems not feasible
- Instead, organizations develop formal evaluation methodologies
 - Products passing evaluation are trusted
 - Required to do business with the organization



TCSEC (83-99): The Original

- Trusted Computer System Evaluation Criteria
 - U.S. Government security evaluation criteria
 - Used for evaluating commercial products
- Policy model based on Bell-LaPadula
 - Emphasis on Confidentiality
- Enforcement: Reference Validation Mechanism
 - Every reference checked by compact, analyzable body of code
- Metric: Seven trust levels:
 - D, C1, C2, B1, B2, B3, A1
 - D is "tried but failed"



Functional Requirements

- Discretionary access control requirements
 - Control sharing of named objects
 - Address propagation of access rights, ACLs, granularity of controls
- Object reuse requirements
 - Hinder attacker gathering information from disk or memory that has been deleted
 - Address overwriting data, revoking access rights, and assignment of resources when data in resource from previous use is present



Functional Requirements

- MAC requirements (B1up)
 - Simple security condition, *-property
 - Description of hierarchy of labels
- Label requirements (B1 up)
 - Used to enforce MAC
 - Address representation of classifications, clearances, exporting labeled information, human-readable output
- Identification, authentication requirements
 - Address granularity of authentication data, protecting that data, associating identity with auditable actions



Functional Requirements

- Audit requirements
 - Define what audit records contain, events to be recorded; set increases as other requirements increase
- Trusted path requirements (B2 up)
 - Communications path guaranteed between user, TCB
- System architecture requirements
 - Tamperproof reference validation mechanism
 - Process isolation
 - Enforcement of principle of least privilege
 - Well-defined user interfaces



Functional Requirements

- Trusted facility management (B2 up)
 - Separation of operator, administrator roles
- Trusted recovery (A1)
 - Securely recover after failure or discontinuity
- System integrity requirement
 - Hardware diagnostics to validate on-site hardware, firmware of TCB



Assurance Requirements

- Configuration management requirements (B2)
 - Identify configuration items, consistent mappings among documentation and code, tools for generating TCB
- System architecture requirements
 - Modularity, minimize complexity, etc.
 - TCB full reference validation mechanism at B3
- Trusted distribution requirement (A1)
 - Address integrity of mapping between masters and on-site versions
 - Address acceptance procedures



Assurance Requirements

- Design specification, verification requirements
 - B1: informal security policy model shown to be consistent with its axioms
 - B2: formal security policy model proven to be consistent with its axioms, descriptive top-level specification (DTLS)
 - B3: DTLS shown to be consistent with security policy model
 - A1: formal top-level specification (FTLS) shown consistent with security policy model using approved formal methods; mapping between FTLS, source code



Assurance Requirements

- Testing requirements
 - Address conformance with claims, resistance to penetration, correction of flaws
 - Requires searching for covert channels for some classes
- Product documentation requirements
 - Security Features User's Guide describes uses, interactions of protection mechanisms
 - Trusted Facility Manual describes requirements for running system securely
- Other documentation: test, design docs



Evaluation Classes A and B

- A1 *Verified protection*; significant use of formal methods; trusted distribution; code, FTLS correspondence
- B3 *Security domains*; full reference validation mechanism; increases trusted path requirements, constrains code development; more DTLS requirements; documentation
- B2 *Structured protection*; formal security policy model; MAC for all objects, labeling; trusted path; least privilege; covert channel analysis, configuration management
- B1 *Labeled security protection*; informal security policy model; MAC for some objects; labeling; more stringent security testing



Evaluation Classes C and D

- C2 *Controlled access protection*; object reuse, auditing, more stringent security testing
- C1 *Discretionary protection*; minimal functional, assurance requirements; I&A controls; DAC
- D Did not meet requirements of any other class



How is Evaluation Done?

- Government-sponsored independent evaluators
 - Application phase:
 - Determine if government cares
 - Preliminary Technical Review phase
 - Discussion of process, schedules
 - Development Process
 - Technical Content, Requirements
 - Evaluation Phase



TCSEC: Evaluation Phases

- Three phases
 - Design analysis
 - Review of design based on documentation
 - Test analysis
 - Final Review
- Trained independent evaluation
 - Results presented to Technical Review Board
 - Must approve before next phase starts
- Ratings Maintenance Program
 - Determines when updates trigger new evaluation



TCSEC: Problems

- Based heavily on confidentiality
 - Did not address integrity, availability
- Tied security and functionality
- Base TCSEC geared to operating systems
 - TNI(87): Trusted Network Interpretation
 - TDI(92): Trusted Database management System Interpretation




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



Later Standards

- CTCPEC (89) – Canada
- ITSEC(91) – European Standard
 - Levels correspond to strength of evaluation (E1—E6, E0)
 - Includes code evaluation, development methodology requirements
 - Introduced Target of Evaluation (TOE), Security target
- CISR: Commercial International Security Req (91)
 - Commercial outgrowth of TCSEC
- Federal Criteria(92): TCSEC Modernization (NIS+NSA)
 - Introduced protection profile (PP)
 - PP: abstract specification of the security aspects of an IT product
- FIPS 140: Cryptographic module validation
- Common Criteria: International Standard
- SSE-CMM: Evaluates developer, not product




ITSEC: Levels

- E1: Security target defined, tested
 - Must have informal architecture description
- E2: Informal description of design
 - Configuration control, distribution control
- E3: Correspondence between code and security target
- E4: Formal model of security policy
 - Structured approach to design
 - Design level vulnerability analysis
- E5: Correspondence between design and code
 - Source code vulnerability analysis
- E6: Formal methods for architecture
 - Formal mapping of design to security policy
 - Mapping of executable to source code



ITSEC Problems:

- No validation that security requirements made sense
 - Product meets goals
 - But does this meet user expectations?
- Inconsistency in evaluations
 - Not as formally defined as TCSEC



FIPS 140: 1994–Present

- Evaluation standard for cryptographic modules (implementing cryptographic logic or processes)
 - Established by US government agencies and Canadian Security Establishment
- Updated in 2001 to address changes in process and technology
 - Officially, FIPS 140-2
- Evaluates only crypto modules
 - If software, processor executing it also included, as is operating system



Requirements

- Four increasing levels of security
- FIPS 140-1 covers basic design, documentation, roles, cryptographic key management, testing, physical security (from electromagnetic interference), etc.
- FIPS 140-2 covers specification, ports & interfaces; finite state model; physical security; mitigation of other attacks; etc.



Security Level 1

- Encryption algorithm must be FIPS-approved algorithm
- Software, firmware components may be executed on general-purpose system using unevaluated OS
- No physical security beyond use of production-grade equipment required



Security Level 2

- More physical security
 - Tamper-proof coatings or seals or pick-resistant locks
- Role-based authentication
 - Module must authenticate that operator is authorized to assume specific role and perform specific services
- Software, firmware components may be executed on multiuser system with OS evaluated at EAL2 or better under Common Criteria
 - Must use one of specified set of protection profiles



Security Level 3

- Enhanced physical security
 - Enough to prevent intruders from accessing critical security parameters within module
- Identity-based authentication
- Stronger requirements for reading, altering critical security parameters
- Software, firmware components require OS to have EAL3 evaluation, trusted path, informal security policy model
 - Can use equivalent evaluated trusted OS instead



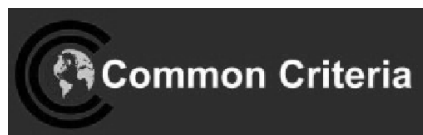
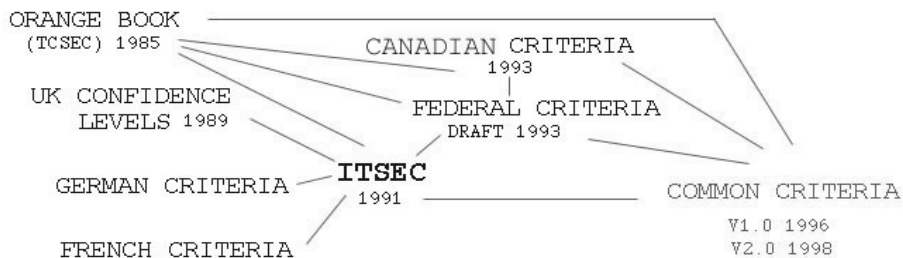
Security Level 4

- “Envelope of protection” around module that detects, responds to all unauthorized attempts at physical access
 - Includes protection against environmental conditions or fluctuations outside module's range of voltage, temperatures
- Software, firmware components require OS meet functional requirements for Security Level 3, and assurance requirements for EAL4
 - Equivalent trusted operating system may be used

Impact

- By 2002, 164 modules, 332 algorithms tested
 - About 50% of modules had security flaws
 - More than 95% of modules had documentation errors
 - About 25% of algorithms had security flaws
 - More than 65% had documentation errors
- Program greatly improved quality, security of cryptographic modules

Common Criteria: 98- Present Origin



- Replaced TCSEC, ITSEC, CTCPEC, FC
- CC had three parts
 1. CC Documents
 - Functional requirements
 - Assurance requirements
 - Evaluation Assurance Levels (EAL)
 2. CC Evaluation Methodology (CEM)
 - Detailed evaluation guidelines for each EAL
 3. National Scheme (Country specific)



Evaluation Methodology

- CC documents
 - Overview of methodology, functional requirements, assurance requirements
- CC Evaluation Methodology (CEM)
 - Detailed guidelines for evaluation at each EAL; currently only EAL1–EAL4 defined
- Evaluation Scheme or National Scheme
 - Country-specific infrastructures implementing CEM
 - In US, it's CC Evaluation and Validation Scheme; NIST accredits commercial labs to do evaluations



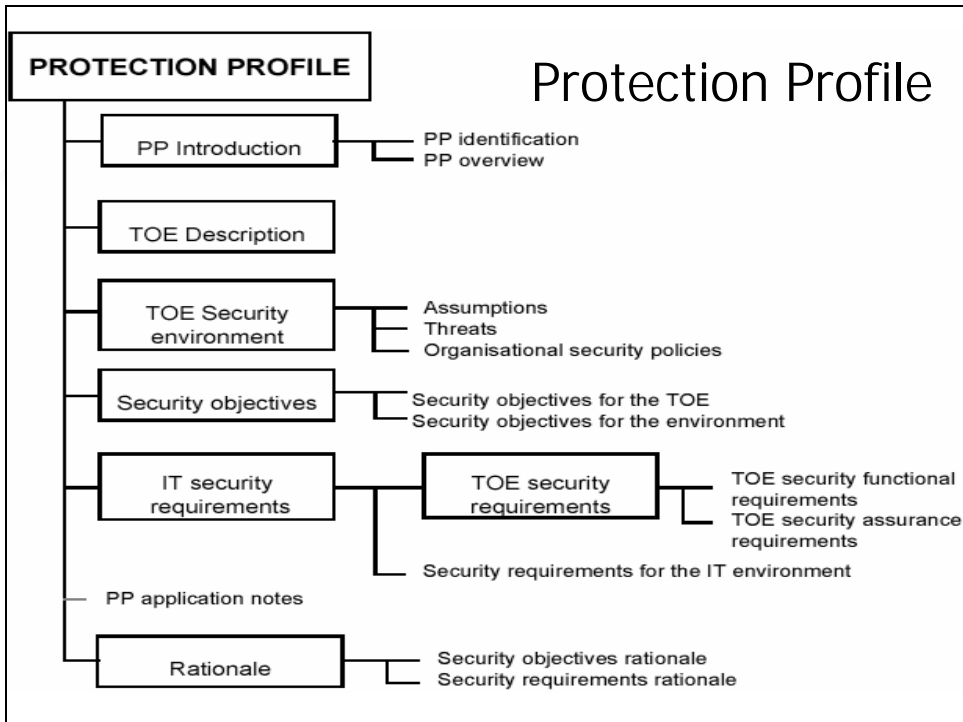
CC Terms

- *Target of Evaluation* (TOE): system or product being evaluated
- *TOE Security Policy* (TSP): set of rules regulating how assets managed, protected, distributed within TOE
- *TOE Security Functions* (TSF): set consisting of all hardware, software, firmware of TOE that must be relied on for correct enforcement of TSP
 - Generalization of TCB



Protection Profiles

- *CC Protection Profile* (PP): implementation-independent set of security requirements for category of products or systems meeting specific consumer needs
 - Includes functional requirements
 - Chosen from CC functional requirements by PP author
 - Includes assurance requirements
 - Chosen from CC assurance requirements; may be EAL plus others
 - PPs for firewalls, desktop systems, etc.
 - Evolved from ideas in earlier criteria



- ## Form of PP
1. Introduction
 - PP Identification and PP Overview
 2. Product or System Family Description
 - Includes description of type, general features of product or system
 3. Product or System Family Security Environment
 - Assumptions about intended use, environment of use;
 - Threats to the assets; and
 - Organizational security policies for product or system



Form of PP (*con't*)

4. Security Objectives
 - Trace security objectives for product back to aspects of identified threats and/or policies
 - Trace security objectives for environment back to threats not completely countered by product or system and/or policies or assumptions not completely met by product or system
5. IT Security Requirements
 - Security functional requirements drawn from CC
 - Security assurance requirements based on an EAL
 - May supply other requirements without reference to CC



Form of PP (*con't*)

6. Rationale
 - Security Objectives Rationale demonstrates stated objectives traceable to all assumptions, threats, policies
 - Security Requirements Rationale demonstrates requirements for product or system and for environment traceable to objectives and meet them
 - This section provides assurance evidence that PP is complete, consistent, technically sound



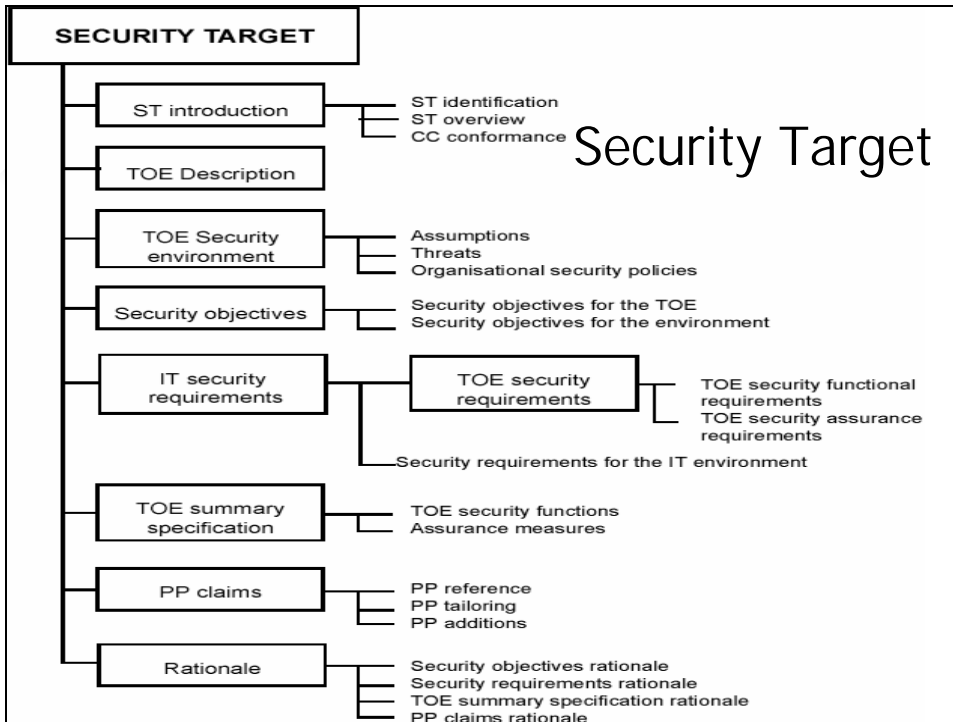
Security Target

- CC Security Target (ST):
 - set of security requirements and specifications to be used as basis for evaluation of identified product or system
 - Can be derived from a PP, or directly from CC
 - If from PP, ST can reference PP directly
 - Addresses issues for *specific* product or system
 - PP addresses issues for a family of potential products or systems



How It Works

- Find appropriate PP and develop appropriate ST based upon it
 - If no PP, use CC to develop ST directly
- Evaluate ST in accordance with assurance class ASE
 - Validates that ST is complete, consistent, technically sound
- Evaluate product or system against ST



- ## Form of ST
1. Introduction
 - ST Identification, ST Overview
 - CC Conformance Claim
 2. Product or System Description
 - Describes TOE as aid to understanding its security requirement
 3. Product or System Family Security Environment
 4. Security Objectives
 5. IT Security Requirements
 - These are the same as for a PP



Form of ST (*con't*)

6. Product or System Summary Specification
 - Statement of security functions, description of how these meet functional requirements
 - Statement of assurance measures specifying how assurance requirements met
7. PP Claims
 - Claims of conformance to (one or more) PP requirements



Form of ST (*con't*)

8. Rationale
 - Security objectives rationale demonstrates stated objectives traceable to assumptions, threats, policies
 - Security requirements rationale demonstrates requirements for TOE and environment traceable to objectives and meets them
 - TOE summary specification rationale demonstrates how TOE security functions and assurance measures meet security requirements
 - Rationale for not meeting all dependencies
 - PP claims rationale explains differences between the ST objectives and requirements and those of any PP to which conformance is claimed



CC Requirements

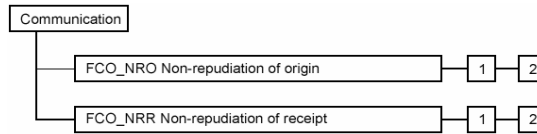
- Both functional and assurance requirements
- EALs built from assurance requirements
- Requirements divided into *classes* based on common purpose
- Classes broken into smaller groups (*families*)
- Families composed of *components*, or sets of definitions of detailed requirements, dependent requirements and definition of hierarchy of requirements



Common Criteria(362 page): Functional Requirements

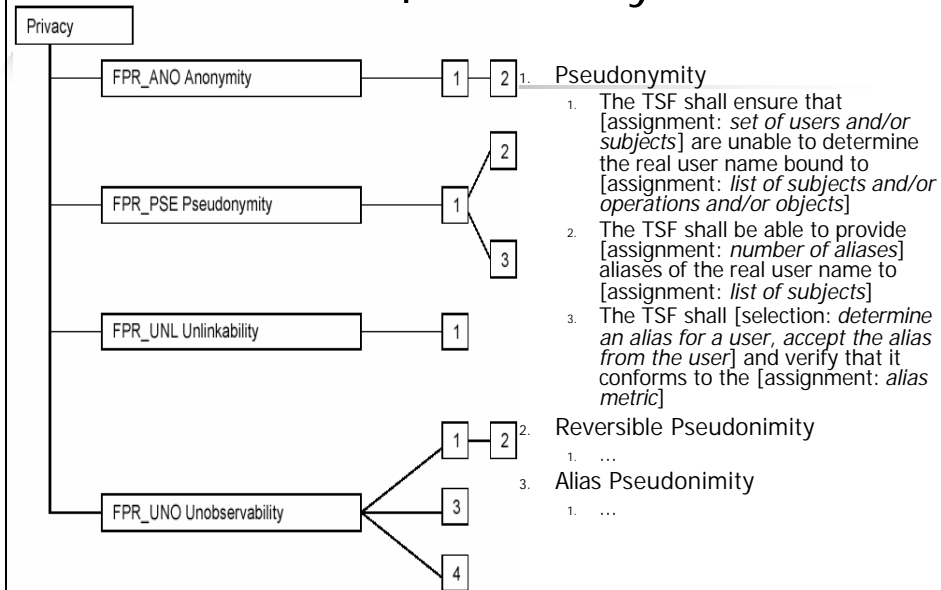
- 11 Classes (Functional requirements)
 - Security Audit (6), Communication (2), Cryptography (2), User data protection (13), ID/authentication (6), Security Management (6), Privacy, Protection of Security Functions (16), Resource Utilization (3), Access (6), Trusted paths (2)
- Several families per class
- Lattice of components in a family

Class Example: Communication



- Non-repudiation of origin
 1. Selective Proof. Capability to request verification of origin
 2. Enforced Proof. All communication includes verifiable origin

Class Example: Privacy



Common Criteria:(216 page) Assurance Requirements

- 10 Classes
 - Protection Profile Evaluation (6), Security Target Evaluation (8), Configuration management (3), Delivery and operation (2), Development (7), Guidance Documentation (2), Life cycle (4), Tests 4), Vulnerability assessment (4), Maintenance of assurance (4)
- Several families per class
- Lattice of components in family

Example: Protection Profile Evaluation

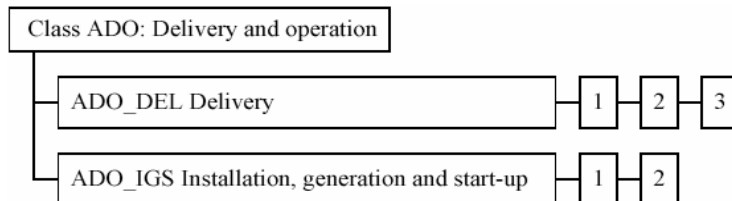
Class APE: Protection Profile evaluation

APE_DES: Protection Profile, TOE description	1
APE_ENV: Protection Profile, Security environment	1
APE_INT: Protection Profile, PP introduction	1
APE_OBJ: Protection Profile, Security objectives	1
APE_REQ: Protection Profile, IT security requirements	1
APE_SRE: Protection Profile, Explicitly stated IT security requirements	1

Security environment

- In order to determine whether the IT security requirements in the PP are sufficient, it is important that the security problem to be solved is clearly understood by all parties to the evaluation.
- 1. Protection Profile, Security environment, Evaluation requirements
 - Dependencies: No dependencies.
 - Developer action elements:
- The PP developer shall provide a statement of TOE security environment as part of the PP.
 - Content and presentation of evidence elements:
-

Example: Delivery and Operation



Installation, generation and start-up

- A. Installation, generation, and start-up procedures
 - Dependencies: AGD_ADM.1 Administrator guidance
- B. Developer action elements:
 - The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
- C. Content and presentation of evidence elements:
 - The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.
- D.

Common Criteria: Evaluation Assurance Levels (EAL)

1. Functionally tested
2. Structurally tested
3. Methodically tested and checked
4. Methodically designed, tested, and reviewed
5. Semi-formally designed and tested
6. Semi-formally verified design and tested
7. Formally verified design and tested

Common Criteria: Evaluation Process

- National Authority authorizes evaluators
 - U.S.: NIST accredits commercial organizations
 - Fee charged for evaluation
- Team of four to six evaluators
 - Develop work plan and clear with NIST
 - Evaluate Protection Profile first
 - If successful, can evaluate Security Target

Common Criteria: Status



- About 80 registered products
 - Only one at level 5 (Java Smart Card)
 - Several OS at 4
 - Likely many more not registered
- New versions appearing on regular basis



National Information Assurance Partnership

Common Criteria Certificate



Microsoft Corporation

The IT product identified in this certificate has been evaluated at an accredited testing laboratory using the Common Methodology for IT Security Evaluation (Version 1.0) for conformance to the Common Criteria for IT Security Evaluation (Version 2.1). This certificate applies only to the specific version and release of the product in its evaluated configuration. The product's functional and assurance security specifications are contained in its security target. The evaluation has been conducted in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence adduced. This certificate is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

Product Name: Windows 2000 Professional, Server, and
Advanced Server with SP3 and Q326886 Hotfix
Evaluation Platform: Compaq Proliant ML570, ML330;
Compaq Professional Workstation AP550; Dell Optiplex
GX400; Dell PE 2500, 6450, 2550, 1550
Assurance Level: EAL4 Augmented

Name of CCTL: Science Applications International
Corporation
Validation Report Number: CCEVS-VR-02-0025
Date Issued: 25 October 2002
Protection Profile Identifier: Controlled Access Protection
Profile, Version 1.d, October 8, 1999

Director
Information Technology Laboratory
National Institute of Standards and Technology

Information Assurance
Director
National Security Agency