Secure OS

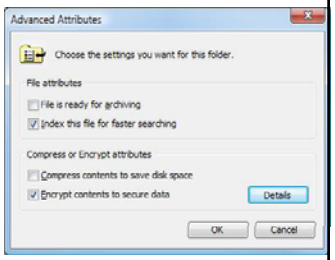
Standard OS Security

Your regular laptop OS provides several security features

- File access permissions
- Separate accounts with passwords
  - Separate home directory
  - Lesser privileged accounts
- Ability to encrypt files transparently

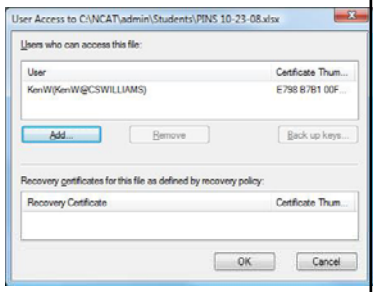
Windows File Encryption

- Windows can encrypt the contents of a file
- Encryption and decryption happens automatically without input from the user
- Access to the encryption keys is based on the user's password

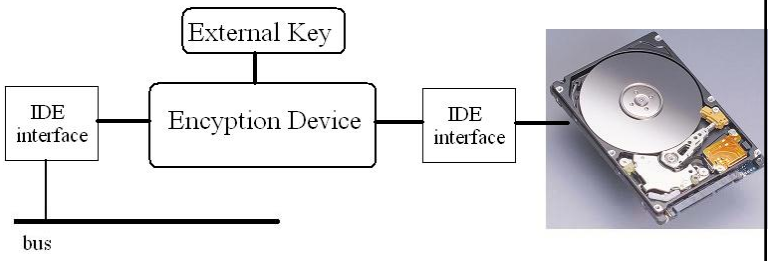


Windows Encryption Recovery

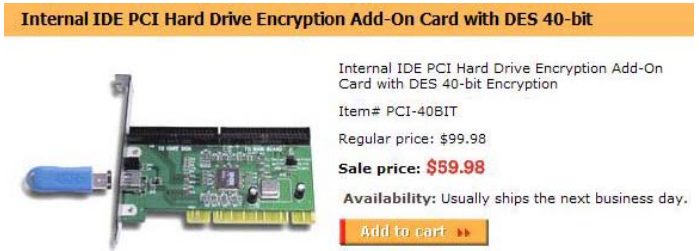
- If an administrator changes a user's password, they will lose access to encrypted files
- Windows supports Recovery Certificates that can allow access if the user previously allowed this



Hardware Disk Encryption



Commercial Products



Windows Security Center

User Account Control

- User Account Control (UAC) asks permission or an admin password before performing actions that could potentially affect your system


Windows needs your permission to continue


Program needs your permission to continue


Unidentified program wants access

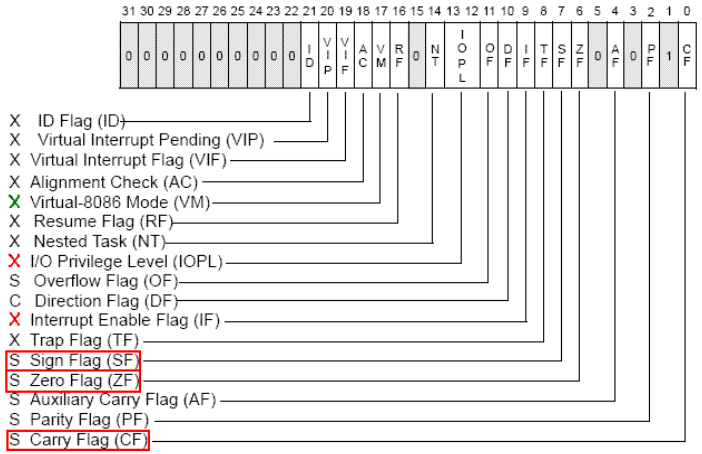

This program has been blocked

- *Many users consider UAC to be annoying*

Pentium Protection

- Protection bits give 4 levels of privilege
  - 0 most protected, 3 least
  - Use of levels software dependent
  - Usually level 3 for applications, level 1 for O/S and level 0 for kernel (level 2 not used)
  - Level 2 may be used for apps that have internal security e.g. database
  - Some instructions only work in level 0

Intel Status Register



Memory Separation

- All memory allocation schemes we discussed prevent one user from accessing the memory of another
- Base address registers have limit registers
- Each user has their own page table in a virtual memory system. Users cannot address the memory of another user.

Security Oriented OS

- There are several operating systems whose focus is enhanced security
- Some OS are designed to enforce mandatory access control policies
- Security-Enhanced Linux (SELinux) is a set of modifications to regular Linux to support access control security policies including mandatory access control

Bell-LaPadula Model

- The Bell-LaPadula Model is a state machine model used for enforcing access control in government and military applications
- Developed by David Elliott Bell and Leonard J. La Padula in 1973
- Basis for the TCSEC and other secure systems

Bell-LaPadula Model, Step 1

- Security levels arranged in linear ordering
  - Top Secret: highest
  - Secret
  - Confidential
  - Unclassified: lowest
- Both people and objects have a *security level*
  - *People or subjects have a clearance level, L*(*s*)
  - Objects have *security classification, L*(*o*)

Marking

- All objects (such as files) in a secure operating system are marked with a security level
- An object's security level is set when the object is created and cannot be changed
- The OS is responsible for maintaining an object's security level
- When a user's account is entered in the system, the administrator sets their security level

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Read rule (Step 1)
  - Subject *s* can read object *o* iff, $L(o) \leq L(s)$ and *s* has permission to read *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- Writing rule (Step 1)
  - Subject *s* can write object *o* iff $L(o) \geq L(s)$ and *s* has permission to write *o*
  - Sometimes called "no writes down" rule

Example

L	*security level*	*subject*	*object*
4	Top Secret	Tanya	Personnel Files
3	Secret	Sam	E-Mail Files
2	Confidential	Claire	Activity Logs
1	Unclassified	Umoja	Telephone Lists

- Tanya can read all files
- Claire cannot read Personnel or E-Mail Files
- Umoja can only read Telephone Lists

Basic Security Theorem, Step 1

- If a system is initially in a secure state, and every action of the system satisfies the read and write rules, then every state of the system is secure
- If all rules are followed, the system is provably secure

Bell-LaPadula Model, Step 2

- Expand notion of security level to include categories
- Security level is (*clearance*, *category set*)
- Examples
  - (Top Secret, { NUC, EUR, ASI })
  - (Confidential, { EUR, ASI })
  - (Secret, { NUC, ASI })

Dominating

- The *dom* relation ("Dominates") specifies that the levels are greater or equal and the categories includes all items in the dominated categories.
- (*L(a)*, *C*) *dom* (*L(b), C´*) iff *L(a)* ≥ *L(b)* and $C' \subseteq C$
- Not a symmetric or asymmetric relation
- Examples
  - (Top Secret, {NUC, ASI}) *dom* (Secret, {NUC})
  - (Secret, {NUC, EUR}) *dom* (Confidential,{NUC, EUR})
  - (Top Secret, {NUC}) ¬*dom* (Confidential, {EUR})
  - (Confidential, {EUR}) ¬*dom* (Top Secret, {NUC})

Levels and Ordering

- Security levels partially ordered
  - Any pair of security levels may (or may not) be related by *dom*
- "dominates" serves the role of "greater than" in step 1
  - "greater than" is a total ordering, though

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Read rule (Step 2)
  - Subject *s* can read object *o* iff *L*(*s*) *dom L*(*o*) and *s* has permission to read *o*
  - Sometimes called "no reads up" rule

Writing Information

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- Write rule (Step 2)
  - Subject *s* can write object *o* iff *L*(*o*) *dom L*(*s*) and *s* has permission to write *o*
  - Sometimes called "no writes down" rule

Examples

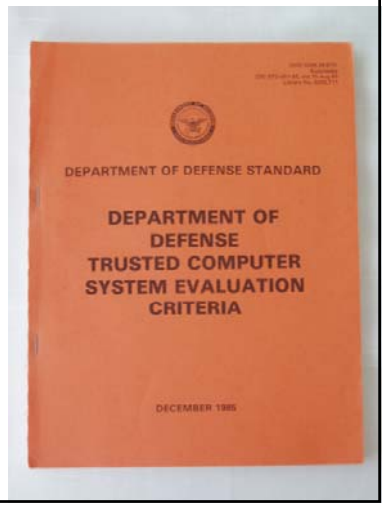
- If Cindy has Top Secret clearance with categories {bombs, encryption} she can read, but not write, a file marked (Secret, {encryption})
- If David has Secret clearance with categories {bombs, encryption} he can read or write a file marked (Secret, {encryption})
- If Amanda has Top Secret clearance with categories {bombs, encryption} she can neither read nor write a file marked (Secret, {covert})

Trusted Computer System Evaluation Criteria

- Trusted Computer System Evaluation Criteria (TCSEC) is a Department of Defense standard that sets basic requirements for assessing the effectiveness of computer security controls
- Created in 1983 and replaced by the international Common Criteria standard in 2005

Orange Book

- The TCSEC standard is informally known as the Orange book
- One of the Rainbow Series of U.S. government security publications



TCSEC Policy Objectives

- The security policy must be explicit, well-defined and enforced by the computer system
- **Mandatory Access Control** – The Bell-LaPadula Model must be enforced
- **Discretionary Security Policy** – Allow users to specify who may access an object and how

TCSEC Accountability Requirements

Regardless of policy the system must enforce

- **Identification** - The ability to recognize an individual user
- **Authentication** - The verification of an individual user's authorization to specific information
- **Auditing** - Audit information must be kept and protected so that actions affecting security can be traced to the individual

TCSEC Assurance

- The computer system must contain hardware/software mechanisms that can be independently evaluated to provide sufficient assurance that the system enforces the above requirements

TCSEC Classes

- **A** — Verified protection
  - A1 — Verified Design
- **B** — Mandatory protection
  - B1 — Labeled Security Protection
  - B2 — Structured Protection
  - B3 — Security Domains
- **C** — Discretionary protection
  - C1 — Discretionary Security Protection
  - C2 — Controlled Access Protection
- **D** — Minimal protection

Common Criteria

- Common Criteria for Information Technology Security Evaluation is an international standard for computer security certification
- Evolved from previous U.S., Canadian and European standards

Evaluation Assurance Level

- The Evaluation Assurance Level (EAL) is a numerical rating describing the depth and rigor of an evaluation
- EAL 1 is the most basic and EAL 7 the most stringent
- Higher EALs do not necessarily imply "better security", only more extensively verified

Common Criteria Testing Laboratories

- Common Criteria Testing Laboratories (CCTL) certify that a product meets specifications
- In the US, the National Institute of Standards and Technology (NIST) and the National Voluntary Laboratory Accreditation Program accredits Laboratories

Why have International Standards?

- If a device has been certified to meet the standard, people building secure systems can feel safe to use it
- Without standards, people can make claims about their security without any basis for accuracy

Criticisms of Common Criteria

- Evaluation is a costly process (often measured in hundreds of thousands of US dollars)
- Evaluation focuses primarily on assessing the documentation, not on the product itself
- It takes so long to prepare the evidence and get certified that the product may be obsolete