INF3510 Information Security
University of Oslo
Spring 2011

Lecture 4

Computer Security

Audun Jøsang

Lecture Overview

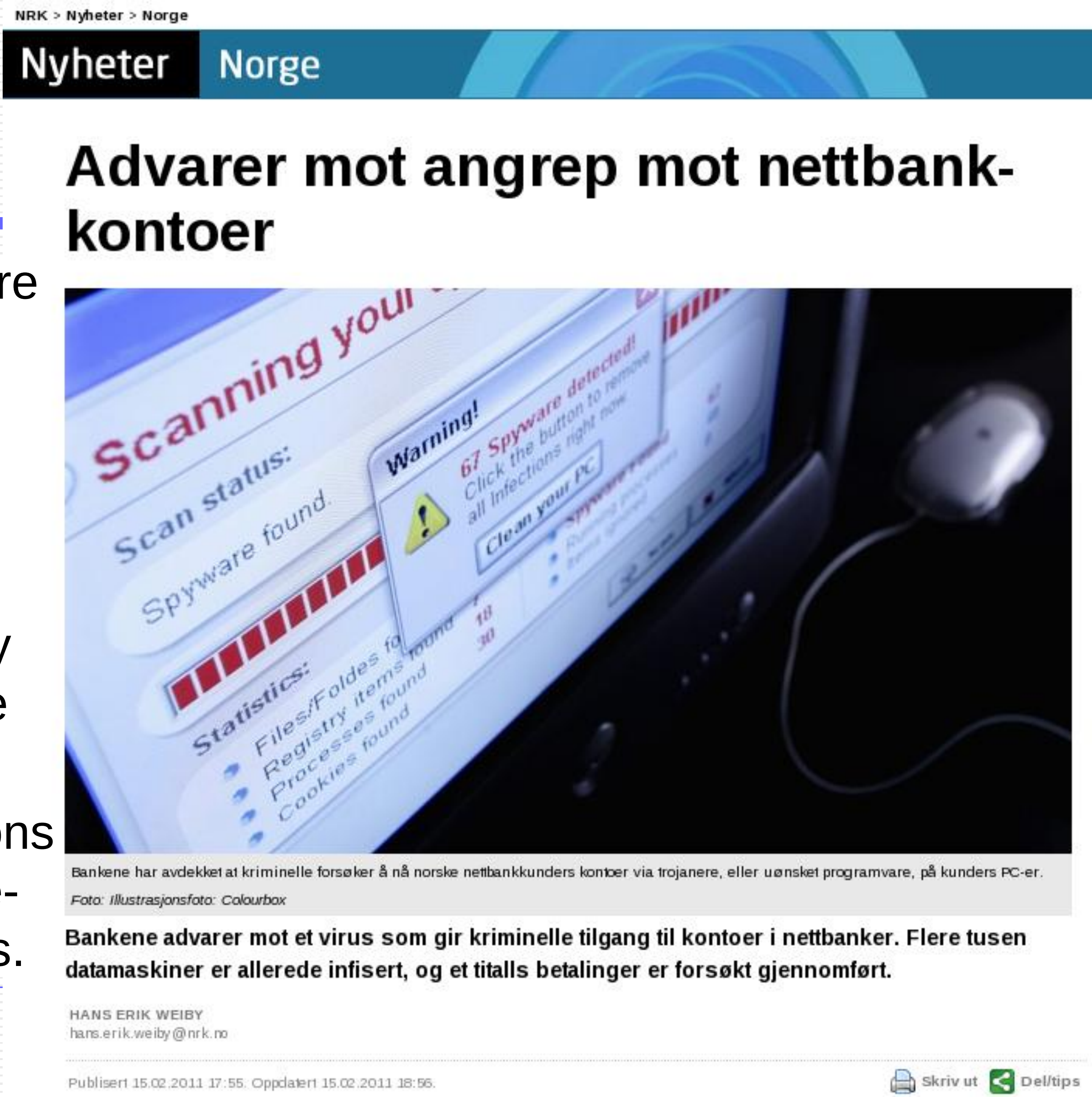
- In the news: Online banking security
- Secure computer architectures
- Trusted computing - background motivation and history
- Security Evaluation

In the news

Online banks are vulnerable to attacks despite strong user authentication and encryption.

PCs infected by ZeuS, SpyeEye Trojans.

Fake transactions with Man-in-the-browser attacks.



NRK > Nyheter > Norge

Nyheter Norge

Advarer mot angrep mot nettbank-kontoer

Bankene har avdekket at kriminelle forsøker å nå norske nettbankkunders kontoer via trojanere, eller uønsket programvare, på kunders PC-er.

Foto: Illustrasjonsfoto: Colourbox

Bankene advarer mot et virus som gir kriminelle tilgang til kontoer i nettbanker. Flere tusen datamaskiner er allerede infisert, og et titalls betalinger er forsøkt gjennomført.

HANS ERIK WEIBY
hans.erik.weiby@nrk.no

Publisert 15.02.2011 17:55. Oppdatert 15.02.2011 18:56.

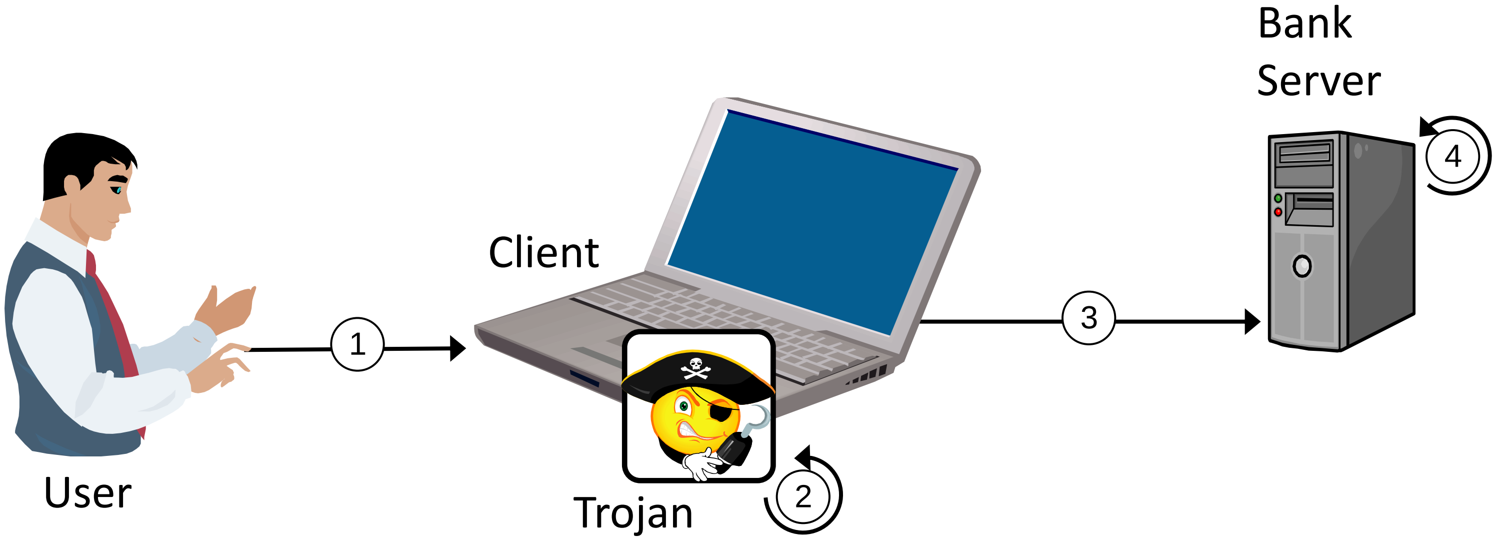
ZeuS and SpyEye Trojans
How they spread

- PCs get exposed to malware attacks when executing programs sent via phishing emails or found on websites.
- Unique attack signatures make AV software ineffective: Viruses evolve their "signature" for each infection or execution, and slip past AV software unnoticed, since AV software is based on known signatures.
- Probably around 50% of PCs are infected with some kind of malware.
- Hackers spend months targeting individual CEOs or businesses to loot their commercial bank accounts or to steal intellectual property.

Man-in-the-browser attacks

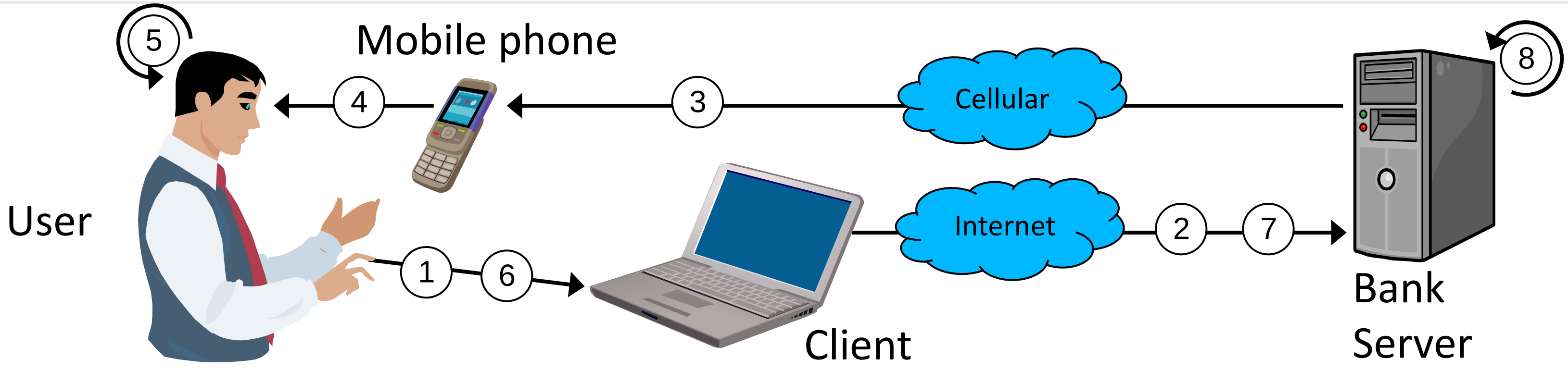
- Malware on a client PC waits until a customer is logged into their bank site, and then spawns a separate hidden window within the session to make fraudulent transactions. Strong user authentication and encryption provide no protection, because attack happens after the user is authenticated and inside the encrypted session.
- ZeuS and SpyEye Trojans: These are not individual viruses but are complete toolkits for sale on the Web. With a full suite of applications and developers around the world adding new capabilities all the time, these programs make it easy for thieves to mount very sophisticated man-in-the-browser attacks and keep them constantly changing.
- The fundamental vulnerability exploited by man-in-the-browser attacks against online banking is the lack of data/transaction authentication. It is not enough to have strong user authentication.

Man-in-the-browser attack scenario



1. Users specifies destination account and amount
2. Trojan changes destination account and amount
3. Transmits wrong transaction with attacker as destination
4. Bank transfers money to attacker

SMS authentication for preventing man-in-the-browser attacks



1. Specify destination account and amount
2. Transaction data transmission
3. SMS with authentication code, destination account and amount
4. View SMS
5. Verify transaction data in SMS
6. If transaction is correct, copy authentication code to browser
7. Transmit authentication code
8. Verify authentication code. If OK, execute transaction.

SMS with transaction details and authentication code

✉ +61412345678

12345678
is your authentication
code from National
Australia Bank for
Funds Transfer of $50
to BSB 123456
Account 123456789

Example mobile
phone SMS message

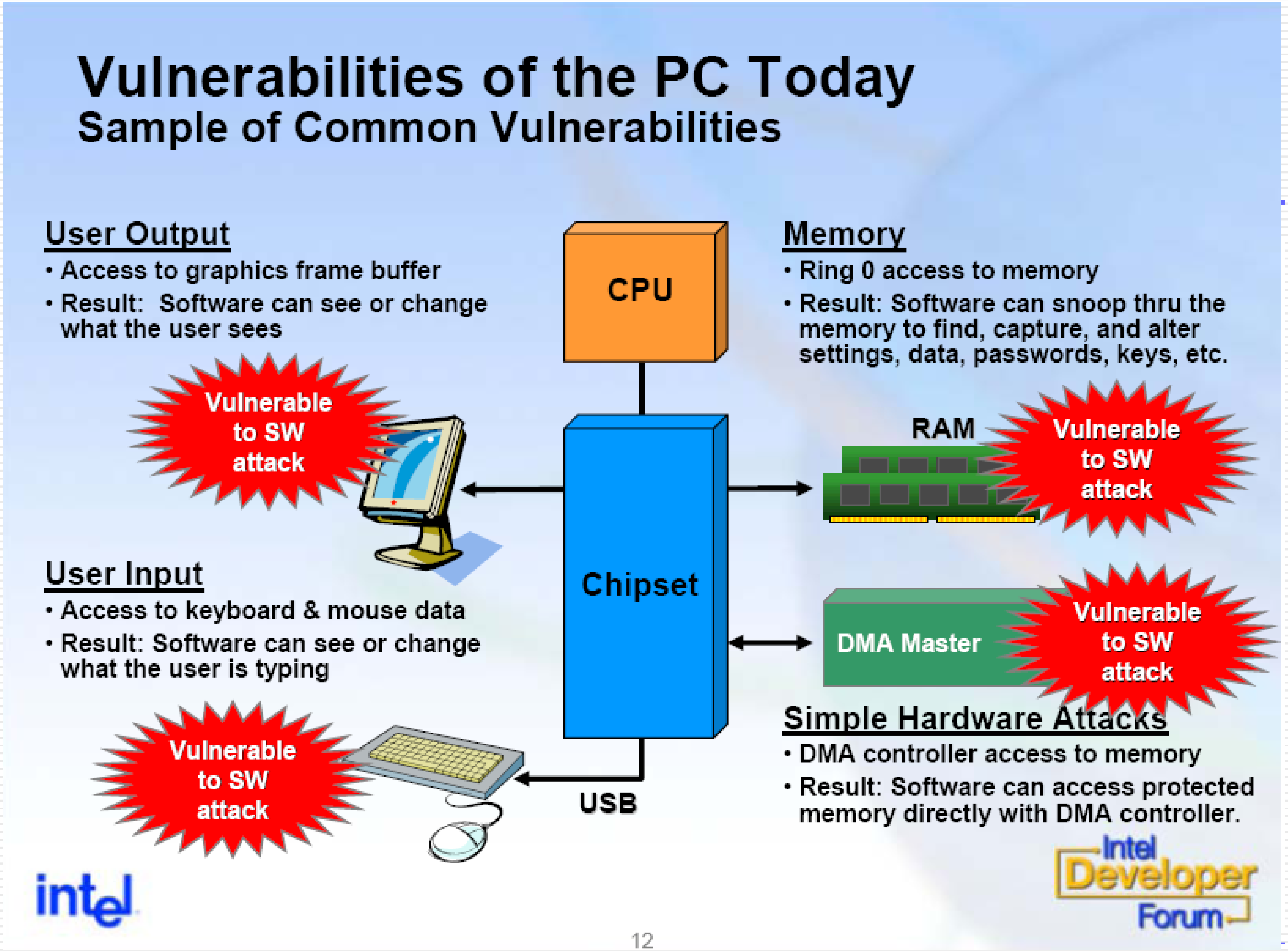
Copy authentication code

Client
terminal

- SMS-based authentication provides verification of transaction data before execution.
- Verifying transaction details in SMS creates a cognitive load which reduces usability.
- With education and awareness this method provides both strong user authentication and strong data/transaction authentication.

Background of computer security

- Increasing reliance on networked computing in commerce and critical infrastructures
- Systems are vulnerable to fraud, vandalism, targeted subversion
- Systems can’t be trusted to operate as expected
- Threats:
  - Subversion via network attacks and mobile code
  - Denial of service
  - ‘Insider’ attacks
  - Application models where the user is motivated to subvert their own device (DRM, software license enforcement, online gaming, electronic cash)



Vulnerabilities of the PC Today

Sample of Common Vulnerabilities

User Output

- Access to graphics frame buffer
- Result: Software can see or change what the user sees

User Input

- Access to keyboard & mouse data
- Result: Software can see or change what the user is typing

Memory

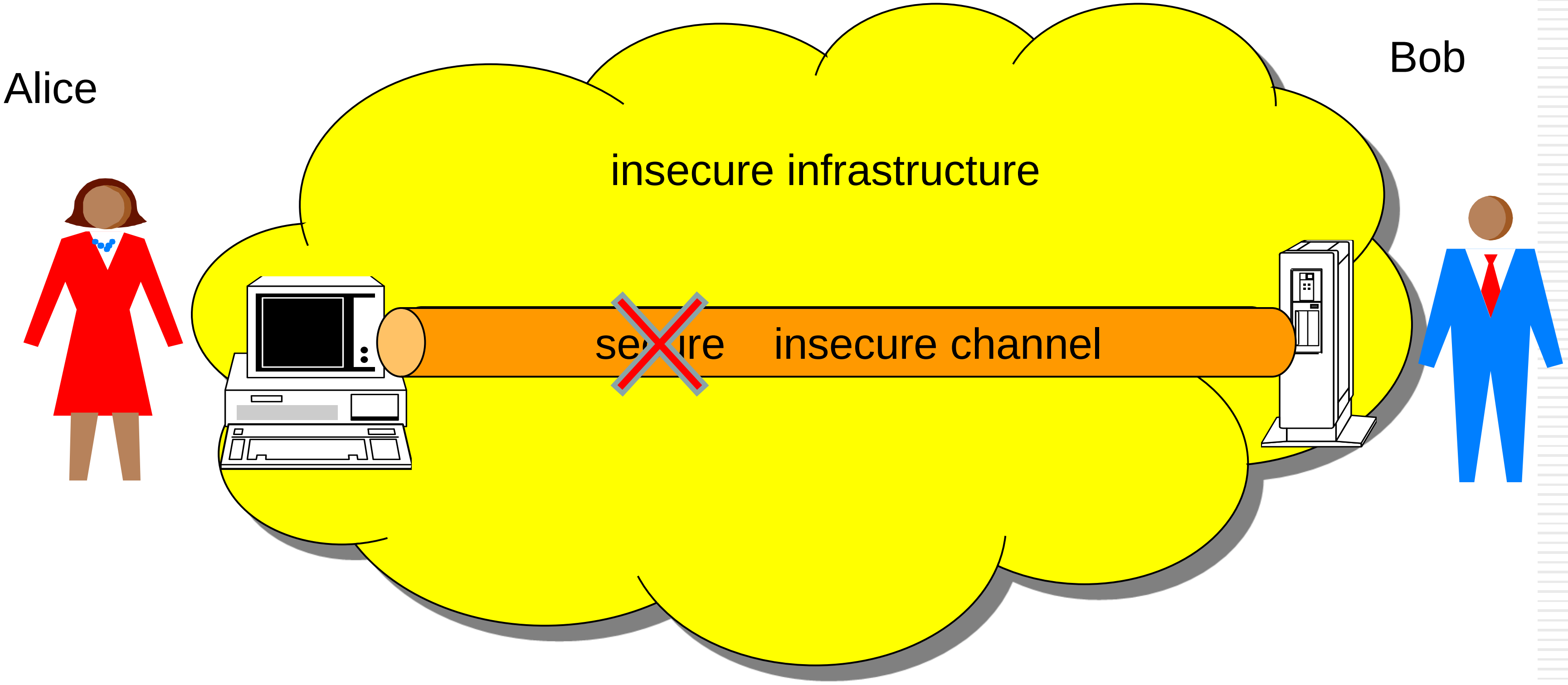
- Ring 0 access to memory
- Result: Software can snoop thru the memory to find, capture, and alter settings, data, passwords, keys, etc.

Simple Hardware Attacks

- DMA controller access to memory
- Result: Software can access protected memory directly with DMA controller.

Source: "LaGrande Architecture" presentation by David Grawrock, delivered at Intel Developer Forum, September 2003. http://www.intel.com/idf/us/fall2003/presentations/F03USSCMS18_OS.pdf

Assumptions and Reality



Approaches to strengthening computer security

- Harden the operating system
  - SE (Security Enhanced) Linux, Trusted Solaris, Windows Vista/7
- Add secure hardware to the commodity platform
  - TPM (Trusted Platform Module)
  - IBM 4764 Secure Coprocessor
- Rely on secure hardware external to the commodity platform
  - Smart cards
  - Hardware tokens
- Give up on the commodity platform (?)

TCB – Trusted Computing Base

- The trusted computing base (TCB) of a computer system is the set of all hardware, firmware, and/or software components that are critical to its security, in the sense that bugs or vulnerabilities occurring inside the TCB might jeopardize the security properties of the entire system.
- By contrast, parts of a computer system outside the TCB must not be able to breach the security policy and may not get any more privileges than are granted to them in accordance to the security policy.

Reference Monitor

- Reference monitor is the specification/description of an access control system which enforces an access control policy over subjects' (e.g., processes and users) ability to perform operations (e.g., read and write) on objects (e.g., files and sockets) on a system.
  - The reference monitor must always be invoked (complete mediation).
  - The reference monitor must be tamperproof (tamperproof).
  - The reference monitor must be small enough to be subject to analysis and tests, the completeness of which can be assured (verifiable).
- The security kernel of an OS is the practical implementation of a reference monitor

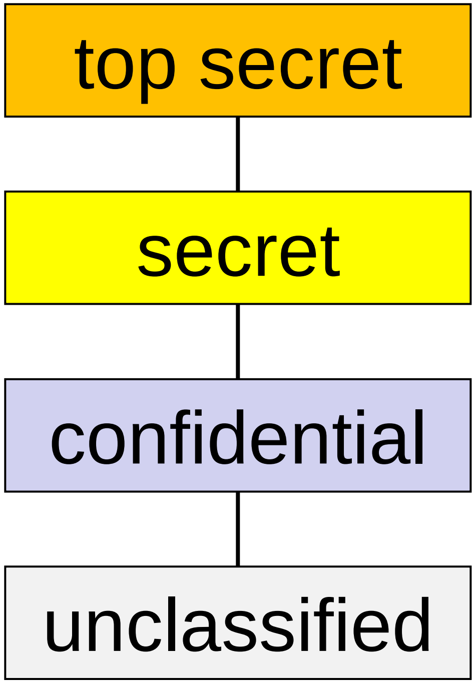
Classical security model: The Bell-LaPadula Model (BLP)

- Probably the most famous security model
- Developed by Bell and La Padula
- Year 1973
- First concerted effort to design a system for enforcing multi-level security in multi-user Operating Systems

BLP and Multilevel Security

- Information is classified according to sensitivity
- Users access information according to their clearance
  - ✓ Subjects and objects are assigned security labels.
  - ✓ No read up
  - ✓ No write-down
  - ✓ + Need to know policy

- Information flows up, not down.



Bell-LaPadula (BLP)

- The idea is to capture the confidentiality aspects of access control
- Access permissions are defined through both:
  - Security levels
  - Access control Matrix (for need-to-know)
- Security policies prevent information flowing downwards from high to low security levels.
- BLP only considers information flow that occurs when a subject observes or alters an object.

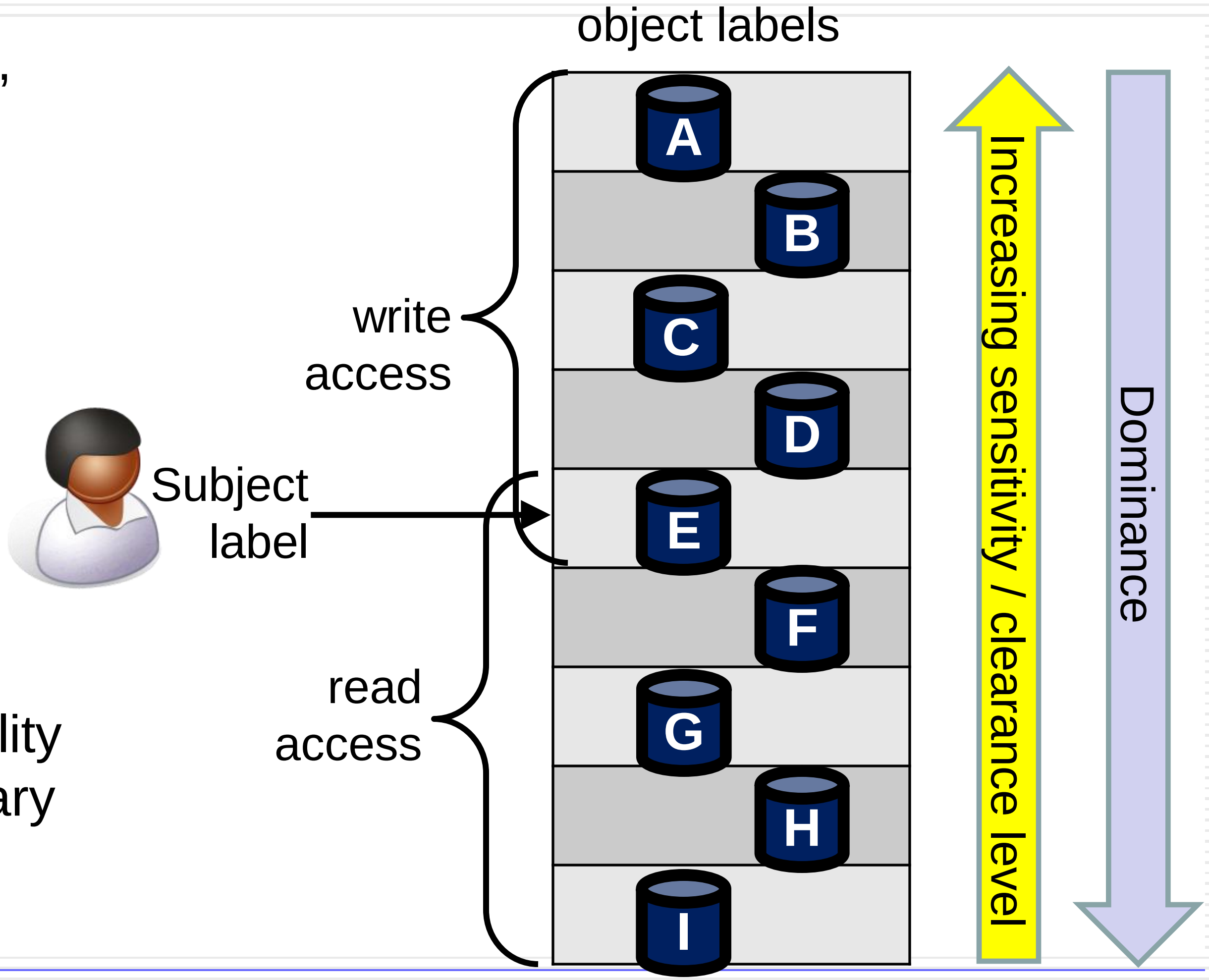
BLP Mandatory Access Control

According to TCSEC, mandatory access control restricts the flow of information according to subject clearance and object sensitivity.

No write down!

No read up!

Typically used for enforcing confidentiality requirements in military settings.

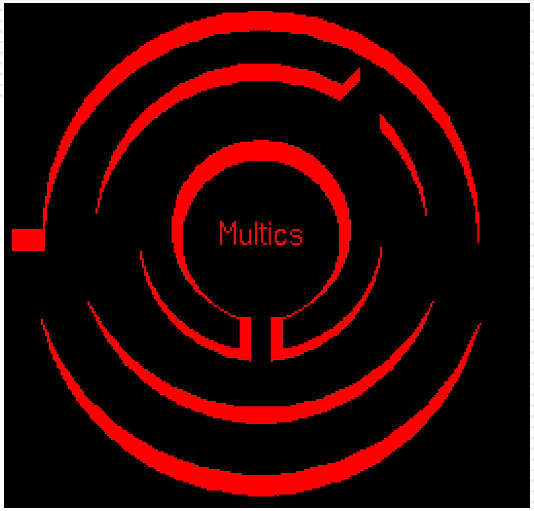


BLP Discretionary Access Control

- Discretionary access control according to TCSEC is defined by an access control matrix, or an ACL (Access Control List)
- Enforces the need-to-know principle

	bob.doc	alice.doc	customers.doc
Alice	-	{read, write}	{read, write}
Bob	{read, write}	{read}	{read}

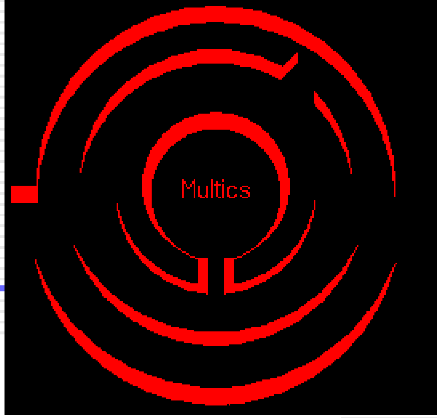
Some history: Multics



- Operating System
  - Designed 1964-1967
    - MIT Project MAC, Bell Labs, GE
  - Introduced timesharing
  - At peak, ~100 Multics sites
  - Last system, Canadian Department of Defense, Nova Scotia, shut down October, 2000
- Extensive Security Mechanisms
  - Security model similar to Bell La Padula
  - First B2 security rating (1980s), the only one for years
  - Influenced many subsequent systems

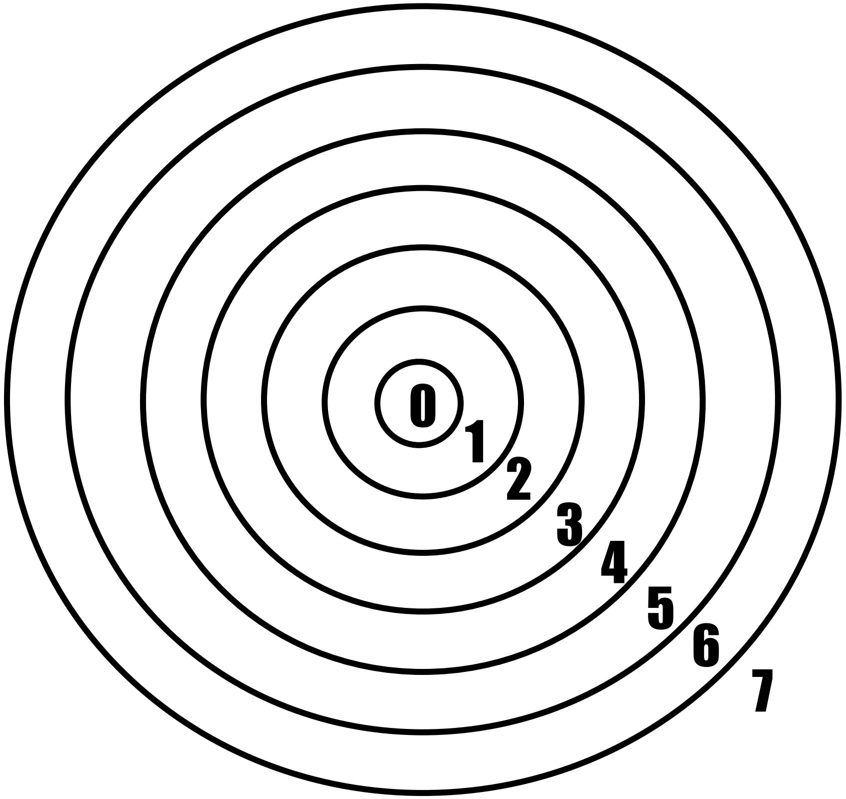
http://www.multicians.org/security.html

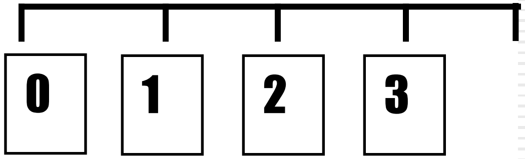
Multics Access Model



- Ring structure
  - A ring is a domain in which a process executes
  - Numbered 0,1, …7 ; Kernel is in ring 0
  - Graduated privileges
    - Processes at ring *i* have privileges of every ring *j* > *i*
- Segments
  - Each data area or procedure is called a segment
  - Segment protection ⟨b1, b2, b3⟩ with b1 ≤ b2 ≤ b3
    - Process/data can be accessed from rings b1 … b2
    - A process from rings b2 … b3 can only call segment at restricted entry points

MULTICS PROTECTION RINGS



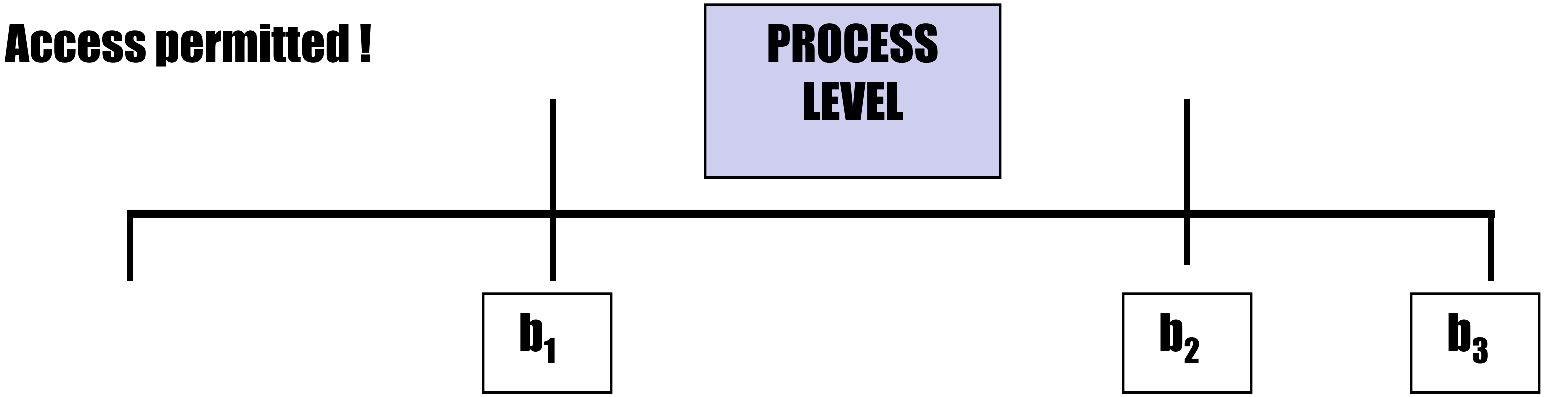


ACCESS BRACKET: b_1 b_2

LIMIT: b_3

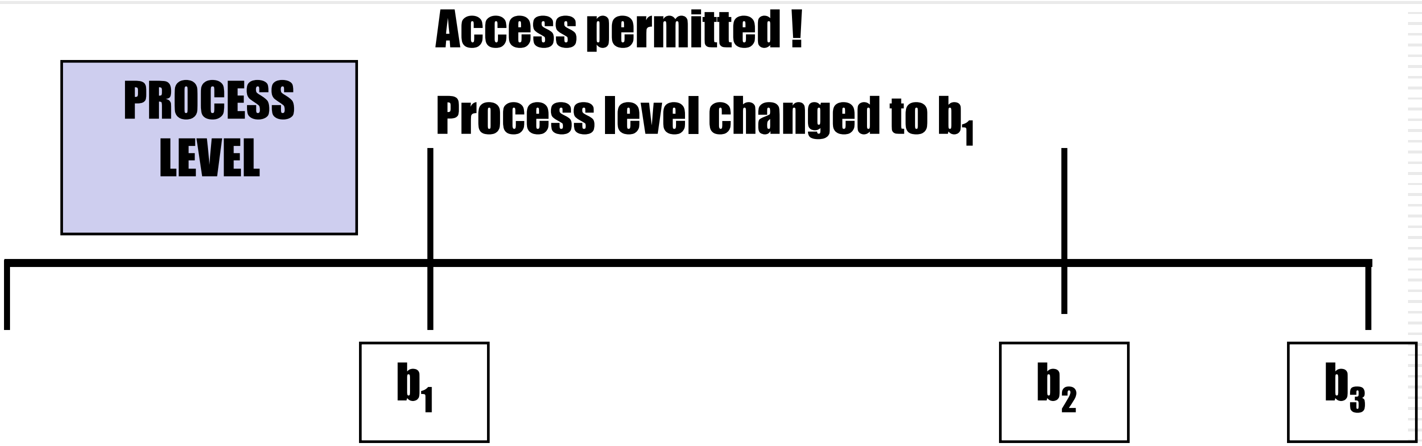
LIST OF GATES: Entry points, at which segments may be called

SEGMENT ACCESS BRACKET



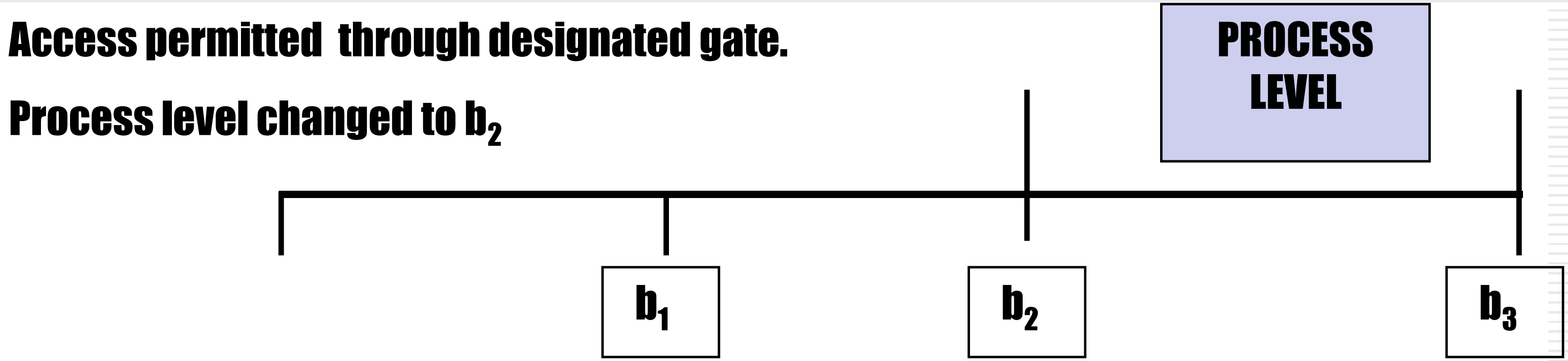
- If a process executing in ring i tries to execute a segment with access bracket (b1, b2), then the call is allowed if b1 <= i <= b2, and the current ring number of the process remained i. Otherwise, a trap to the kernel occurrs.

SEGMENT ACCESS BRACKET



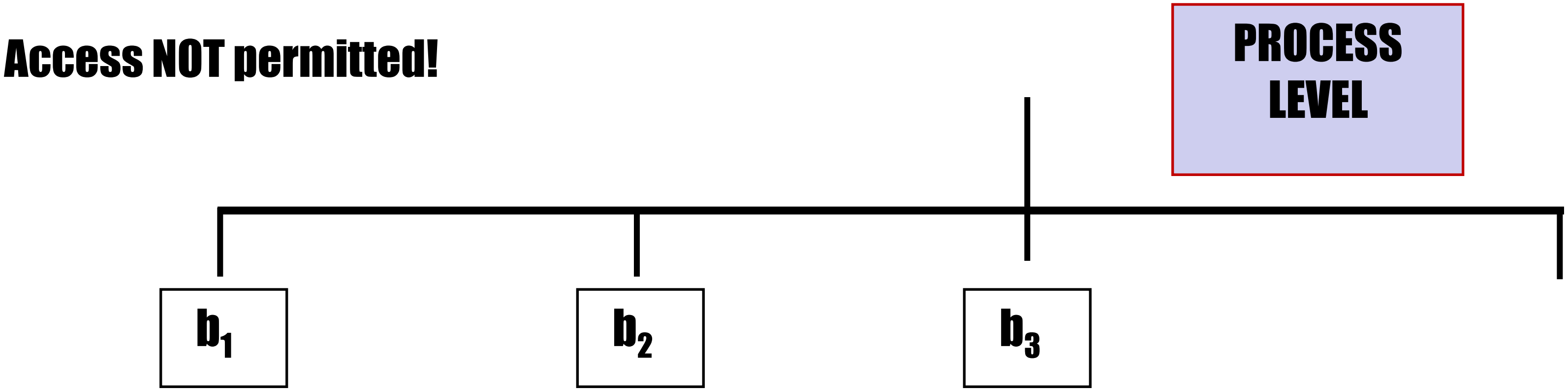
- If i <= b1, then the call is allowed to occur and the current-ring-no of the process is changed to b1. Thus the access rights of the process are reduced. If parameters are passed which refer to segments in a ring lower than b1, then these segments were copied into an area accessible in ring b1.

SEGMENT ACCESS BRACKET


Access permitted through designated gate.
Process level changed to b_2
PROCESS LEVEL
b_1
b_2
b_3

- If i > b2, then the call is allowed to occur only if i <= b3, and the call is directed to one of the designated OS call gates. If successful, the current-ring-nr of the process is changed to b2.

SEGMENT ACCESS BRACKET



- If i > b3 (the limit) no access is permitted.

MODERN SYSTEMS

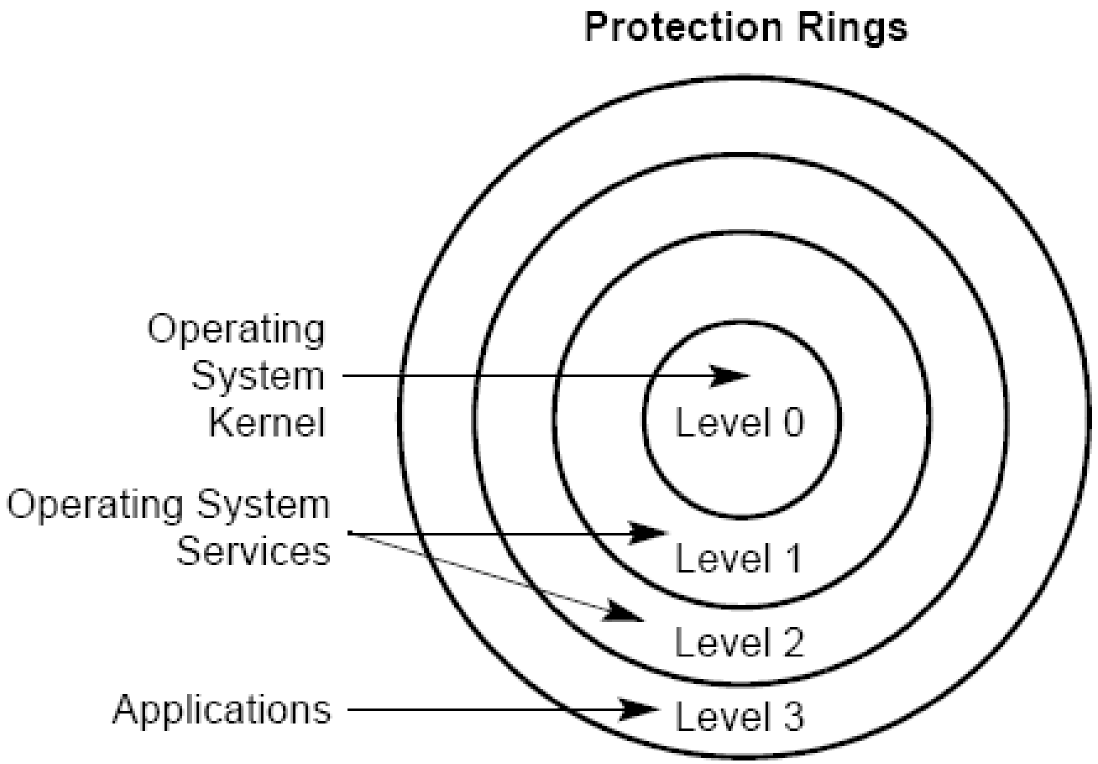
- Intel x86 processors have 4 privilege levels with the intended use as follows

– Ring 0:	kernel
– Ring 1 & 2:	device drivers
– Ring 3:	applications

- Documentation:
  - Intel64 and IA-32 Architectures Software Developer’s Manual, Volume 3A, Chapter 5
  - http://www.intel.com/design/processor/manuals/253668.pdf

Intel Memory Protection Rings

- Originally in Multics OS Software
- In Intel hardware architecture since 80386



Execution Privilege Levels

- CPU enforces constraints on memory access and changes of control between different privilege levels
- Similar in spirit to Bell-LaPadula access control restrictions
- Hardware enforcement of division between user mode and kernel mode in operating systems
  - Simple malicious code cannot jump into kernel space

Privileged Instructions

Some of the system instructions (called “privileged instructions”) are protected from use by application programs. The privileged instructions control system functions (such as the loading of system registers). They can be executed only when the Privilege Level is 0 (most privileged). If one of these instructions is executed when the Privilege Level is not 0, general-protection exception (#GP) is generated, and the program crashes.

WHAT ABOUT DRIVERS

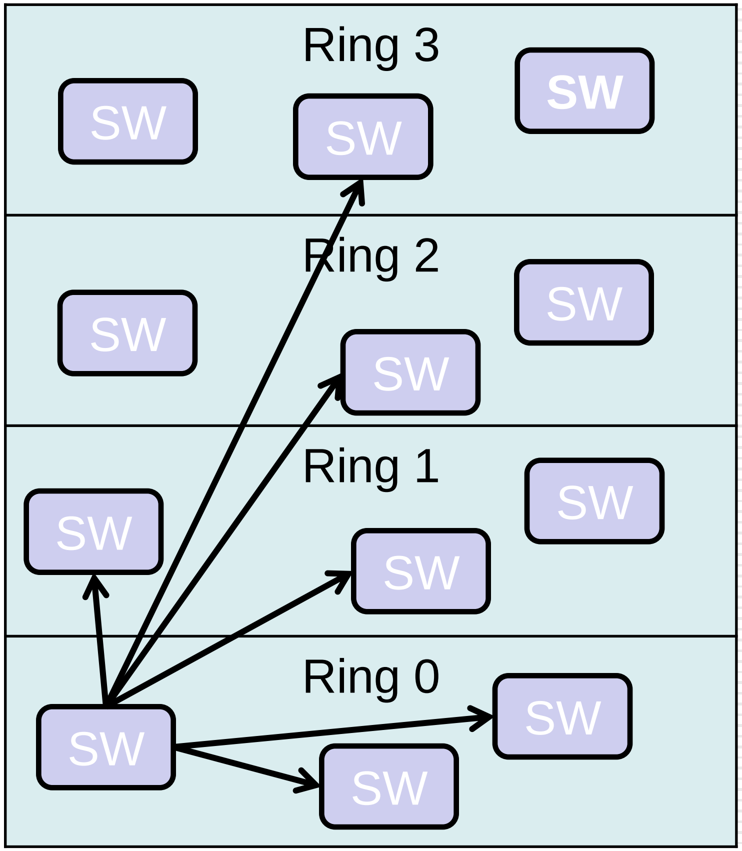
- *Can somebody please tell me why a fault in my sound card driver has to crash my system?*
- Answer: MS Windows only uses ring 0 and 3.
  - Windows 98 had device drivers in ring 3. Execution became very slow because driver access had to go via OS calls.
  - From Windows 2000, they are in ring 0, for performance reasons

Limiting Memory Access Type

- The Pentium architecture supports making pages read/only versus read/write
- A recent development is the Execute Disable Bit
  - Added in 2001 but only available in systems recently
  - Supported by Windows XP SP2 an later
- Similar functionality in AMD Altheon 64
  - Called Enhanced Virus Protection

Robustness of protection ring model

- A process can access and modify any data and software at the same or less privileged level as itself.
- A process that runs in kernel mode (ring 0) can thus modify anything on the whole platform.
- The goal of attackers is to get access to kernel mode.
  - through exploits
  - by tricking users to install software



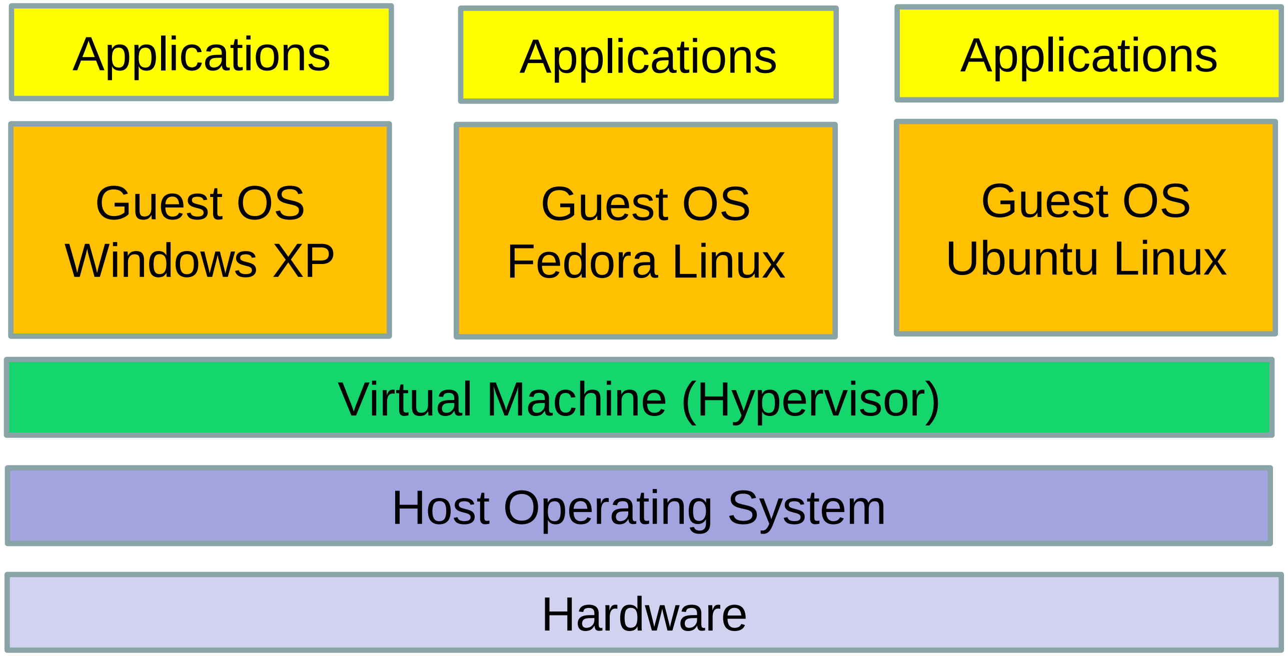
Virtual Machine

- A software implementation of a machine (computer)
- that executes programs like a real machine.
- • Example: Java Virtual Machine (JVM)
- – JVM accepts a form of computer intermediate language
- commonly referred to as Jave bytecode.
- • "compile once, run anywhere“
- – The JVM translates the bytecode to executable instructions
- on the fly

Platform Virtualization

- Allows one or more Operating System to
- execute on top of another Operating System
- There are lots of VM-software available
  - VMWare is probably the most known
    - Commercial product
    - Free version comes with a limitations
  - VirtualBox is a software for x86 virtualization
    - It is freely availably under GPL
    - Runs on Windows, Linux, OS X and Solaris hosts

Virtual Machine Architecture



Why run a virtual machine?

- Run several OS at the same time
- Take a snapshot of the current state of the OS
  - Use this later on to reset the system to that state
- Example of use
  - Testing
  - Malware Analysis
  - Keep several servers running on one physical machine (green IT hype)

Buffer overflow

- A program tries to store more data in a buffer than it was intended to hold.
- Example:
  - We have a 5 bytes buffer in memory:

  - We fill it with 10 bytes so that 5 extra bytes get overwritten

  - When the overwritten part contains software, it is possible to change that software to do something the attacker wants.
- Many attacks use buffer overflow techniques

Trusted Computing

Trusted Computing Motivation

- Computer Security
  - Well established since 1960s
- Trusted Computing Base (TCB)
  - The totality of protection mechanisms within a computer system, including hardware, firmware and software
  - Concept developed during 1980s
- Physical access to computers open up for attacks that can circumvent traditional TCBs, e.g. secure operating systems
- Complexity of contemporary systems makes it impossible to remove all software vulnerabilities

Basic idea of Trusted Computing

- Addition of security hardware functionality to a computer system
- Enables external entities to have increased level of trust that the system will perform as expected/specified

Related Concept: Trusted Platform

- Trusted platform = a computing platform with a secure hardware component that forms a security foundation for software processes
- Trusted Computing = computing on a Trusted Platform

Motivation for Trusted Hardware

- Computing platforms are deployed in hostile environments , in contrast to 1960’s 1970’s protected computing centres
  - There is a gap between the reality of physically unprotected, network connected systems and the assumption of confidentiality and integrity
  - The gap must be closed if systems are to be trustworthy

What is “trust” in the sense of TC?

- To have faith or confidence that something desired is, or will be, the case
- Trust engenders confident expectations
- Trust allows us to believe assertions

  “*A trusted component, operation, or process is one whose behaviour is predictable under almost any operating condition and which is highly resistant to subversion by application software, viruses, and a given level of physical interference*”
- A ‘trusted’ component can violate the security policy if it breaks
- A ‘trustworthy’ component can be relied on to enforce the security policy, because it doesn’t break
- A ‘trusted system’ can be verified to enforce a given security policy
- The big question: “Trusted by whom to do what?”

Trusted by whom to do what?

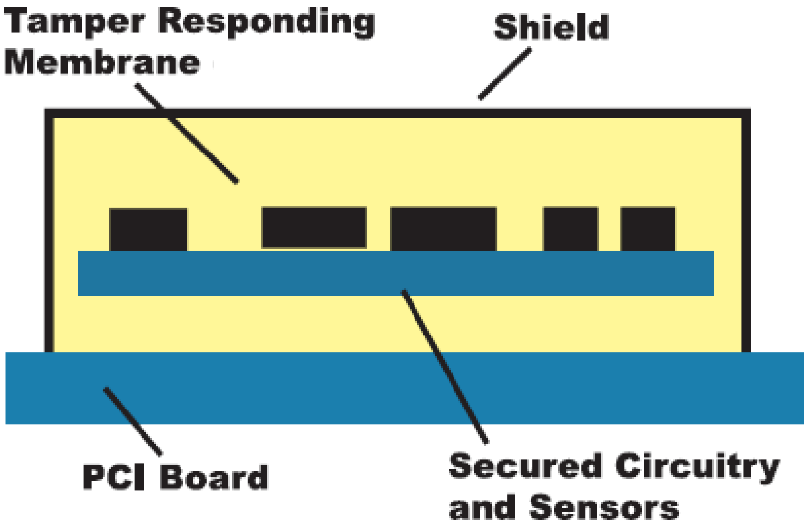
- Has the OS been subverted?
  - Virus/Trojan/Spyware/Rootkit
  - Keystroke/screen/mouse logger
  - Smart card reader, biometric reader access
- How would the user know?
- How would a program on another computer know?

Characteristics of Trusted Hardware

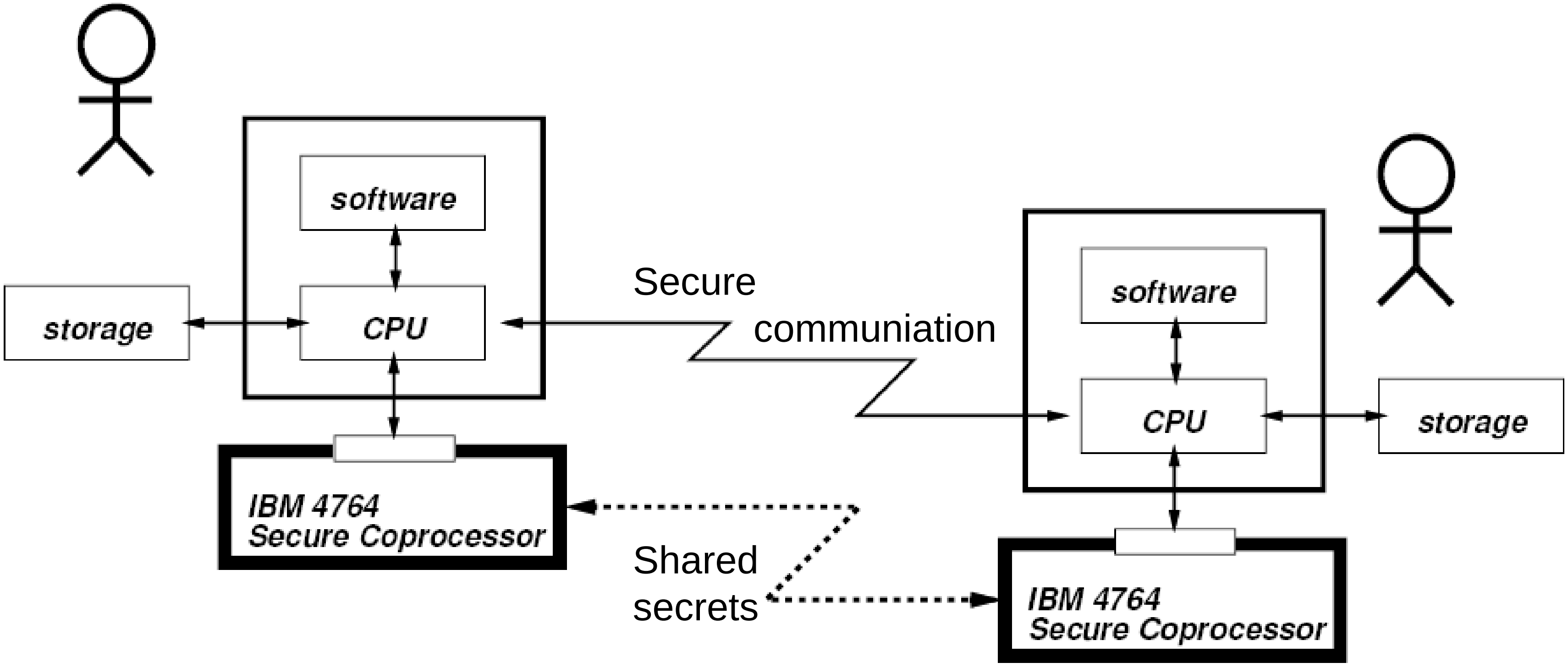
- Physically secure module
- Environmental monitoring (temperature, power supply, structural integrity)
- Tamper responsive
- CPU
- ROM for OS and application code
- NVRAM (Flash), EEPROM, BBRAM for secrets and data (zeroisation)
- Optimized hardware support for cryptography
- I/O interface

Trusted Hardware – Example

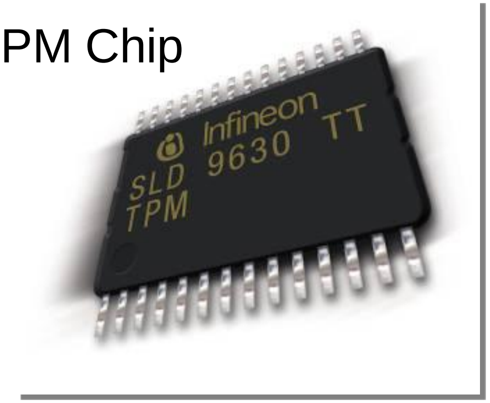
- IBM 4764 Secure Coprocessor



IBM 4764 Application Example



Trusted Hardware Examples



TPM Chip

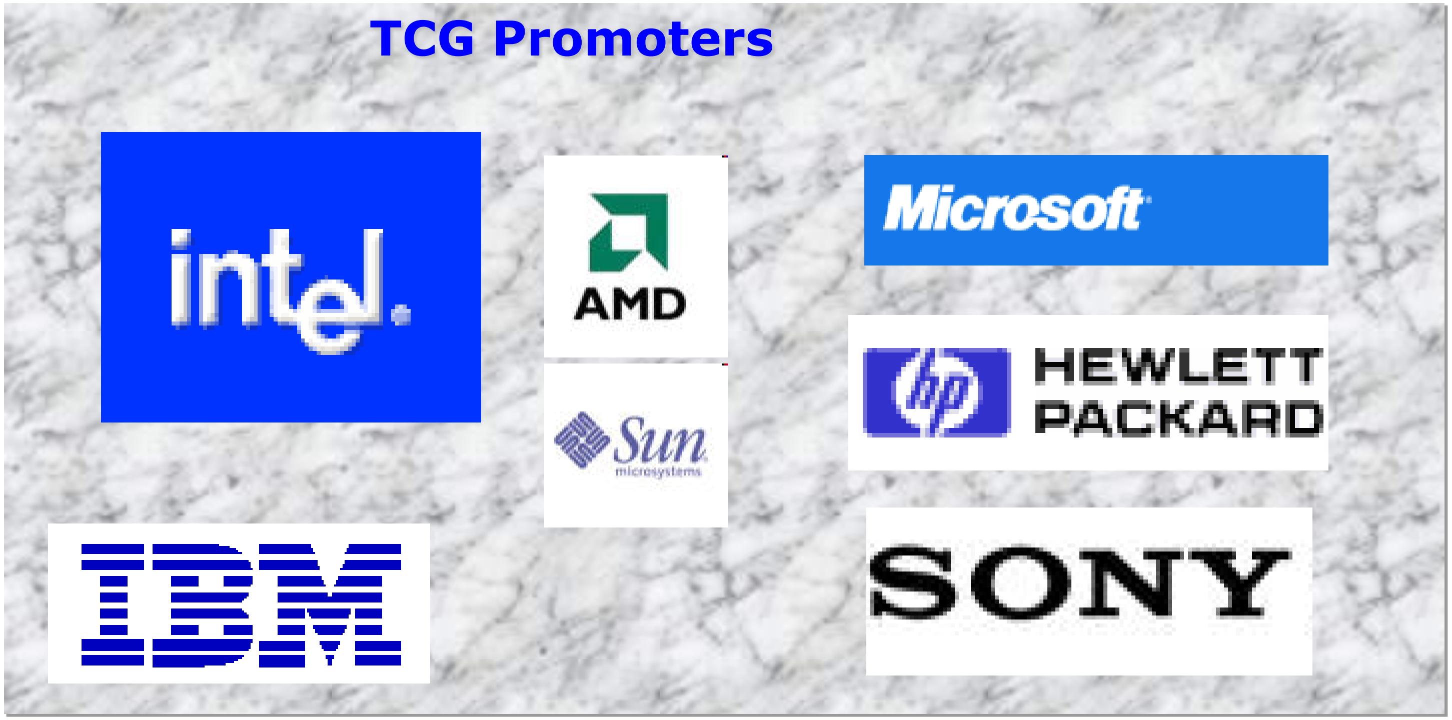
iButton

IBM 4764

Fortezza PC Card

Smart Card

Trusted Computing Group (TCG)



TCG History & Evolution

- October 1999: TCPA formed
  - Trusted Computing Platform Alliance
  - Founders: IBM, HP, Compaq, Intel and Microsoft
- 2001: 1st TPM specification released
  - Trusted Platform Module
- 2002: TCPA becomes TCG
  - Trusted Computing Group
  - Incorporated not-for-profit industry standards organization
- 2003: TCPA TPM specification adopted by TCG
  - Currently TPM specification 1.2

Trusted Platform Module (TPM)

- Hardware module at heart of hardware / software approach to trusted computing
- Protected memory (key storage, platform configuration metrics)
- TPM chip mounted on motherboard,
- Supports 3 basic services:
  - Secure / authenticated boot,
  - Remote attestation, allows remote party to verify platform state
  - Sealed storage / encryption, makes decryption depend on platform state

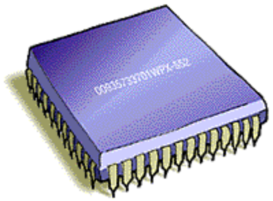
TCG supports two modes of booting

- Secure boot
  - the platform owner can define expected (trusted) PCR values that are stored in special non-volatile Data Integrity Registers (DIR) in the TPM.
  - If a PCR value does not match the expected value for that stage of the boot process, TPM can signal a boot termination request.
- Authenticated boot
  - does not check measured values against expected values – just records in PCRs

TPM – A Passive Security Enabler

- Note that TPM is passive:
  - It doesn’t *decide* which software can and can’t run.
  - It provides a way to reliably report the post-boot state of the platform
  - TCG aware *application or OS* can be designed to not start unless platform is in a particular state (no malware etc)
  - TCG aware *application or OS* can be designed to require a TPM mediated online authorisation from a vendor before starting (check for current license etc.):
    - TCG can be *used* to build systems where somebody else decides whether software can or can’t run

      TCG does not provide this functionality – it merely enables it



Microsoft Vista & Windows 7 BitLocker

- Disk volume encryption
- Off-line protection only
- Protects against data loss in case of lost/stolen computers
- Can be based on TPM, but not necessarily

Spectrum of Protection

BitLocker offers different types of protection, depending on needs

TPM + USB

"What it is + what you have"

Protects Against:
HW attacks

Vulnerable To:
Stolen USB key

User Must:
Protect USB key

USB Only

"What you have"

Protects Against:
HW attacks

Vulnerable To:
Stolen USB key
No boot validation

User Must:
Protect USB key

TPM + PIN

****

"What it is + what you know"

Protects Against:
Many HW attacks

Vulnerable To:
Hardware attacks

User Must:
Enter PIN to boot

TPM Only

"What it is"

Protects Against:
Most SW attacks

Vulnerable To:
Hardware attacks

User Must:
N/A
No user impact

Ease of Deployment / Maintenance

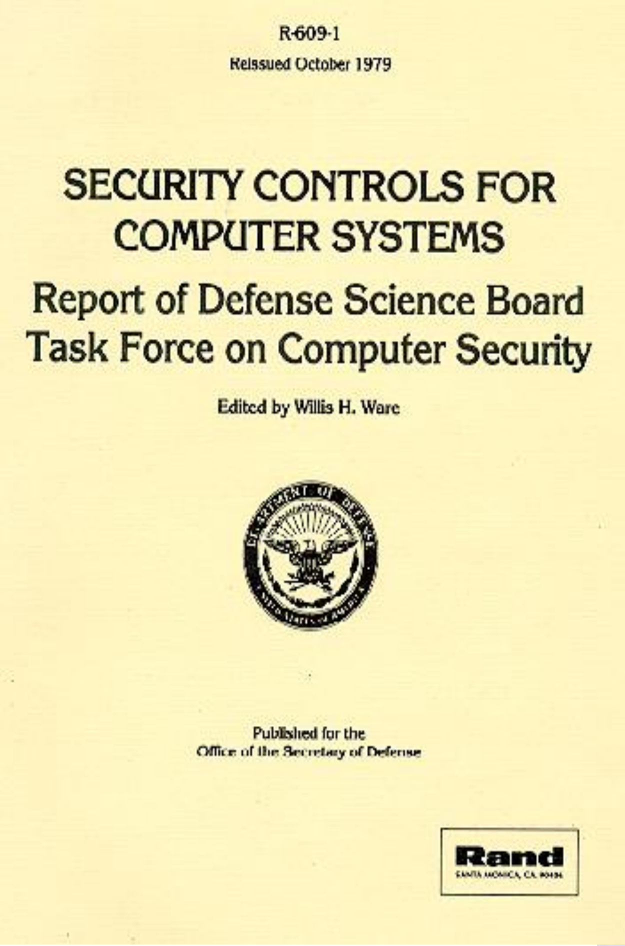
BitLocker life Cycle

- Installation
  - Select protection
  - Select recovery password or key
- Operation - 4 different modes:
  - TPM only, TPM+PIN, TPM+USB, USB only
- Decommissioning
  - Remove keys by formatting volume
  - Remove BitLocker key protectors
  - Reset TPM

Security Evaluation

Security Evaluation

- How do you get assurance that your computer systems are adequately secure?
- You could trust your software providers.
- You could check the software yourself, but you would have to be a real expert, and it would take long.
- You could rely on an impartial security evaluation by an independent body.
- Security evaluation schemes have evolved since the 1980s; currently the Common Criteria are used internationally.



R-609-1
Reissued October 1979

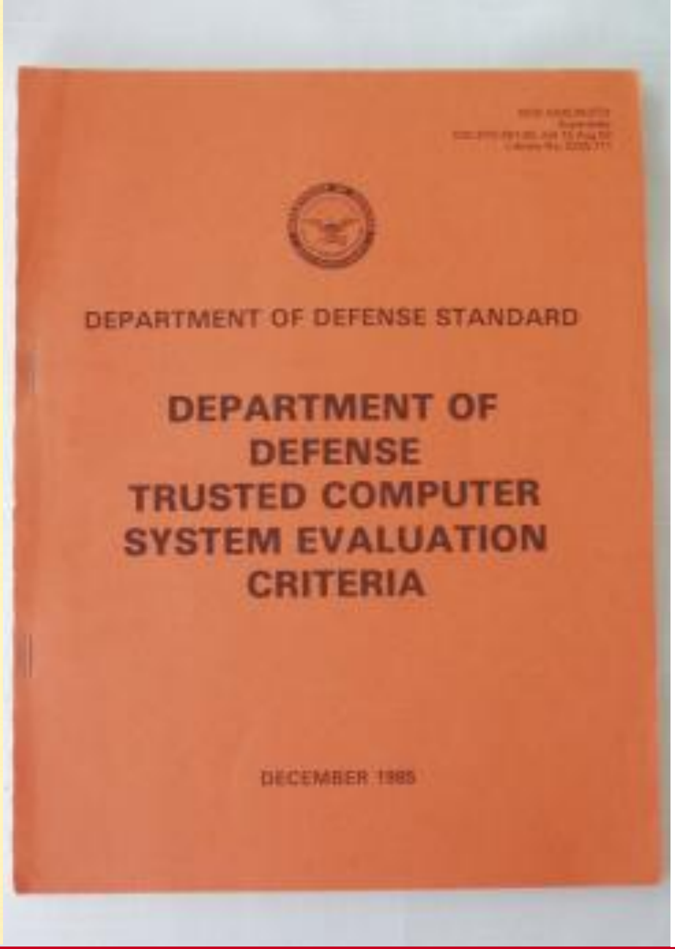
SECURITY CONTROLS FOR COMPUTER SYSTEMS

Report of Defense Science Board Task Force on Computer Security

Edited by Willis H. Ware

Published for the
Office of the Secretary of Defense

Rand
SANTA MONICA, CA 90406

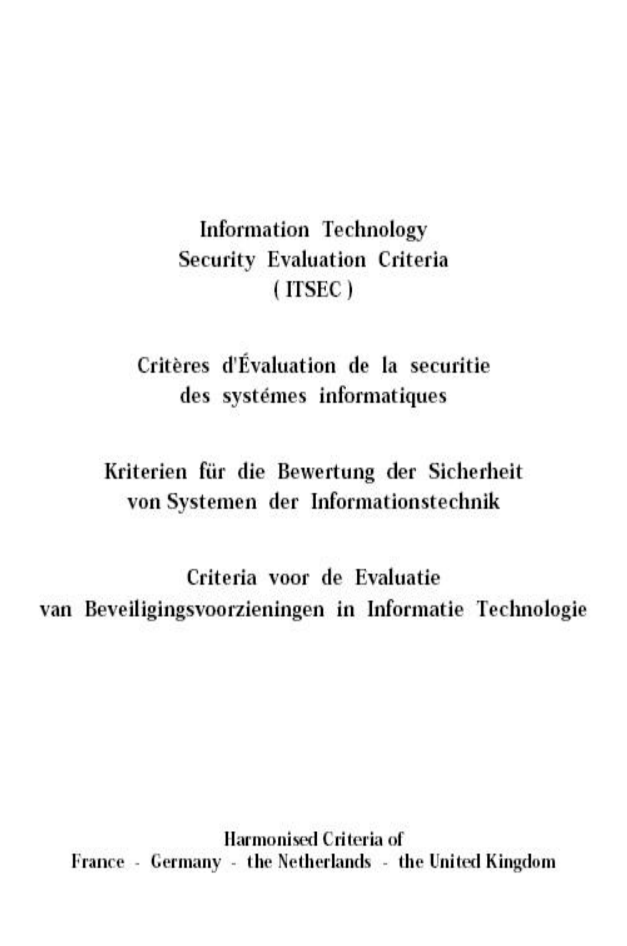


[illegible]

DEPARTMENT OF DEFENSE STANDARD

DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA

DECEMBER 1985



Information Technology
Security Evaluation Criteria
(ITSEC)

Critères d'Évaluation de la securitie
des systémes informatiques

Kriterien für die Bewertung der Sicherheit
von Systemen der Informationstechnik

Criteria voor de Evaluatie
van Beveiligingsvoorzieningen in Informatie Technologie

Harmonised Criteria of
France - Germany - the Netherlands - the United Kingdom

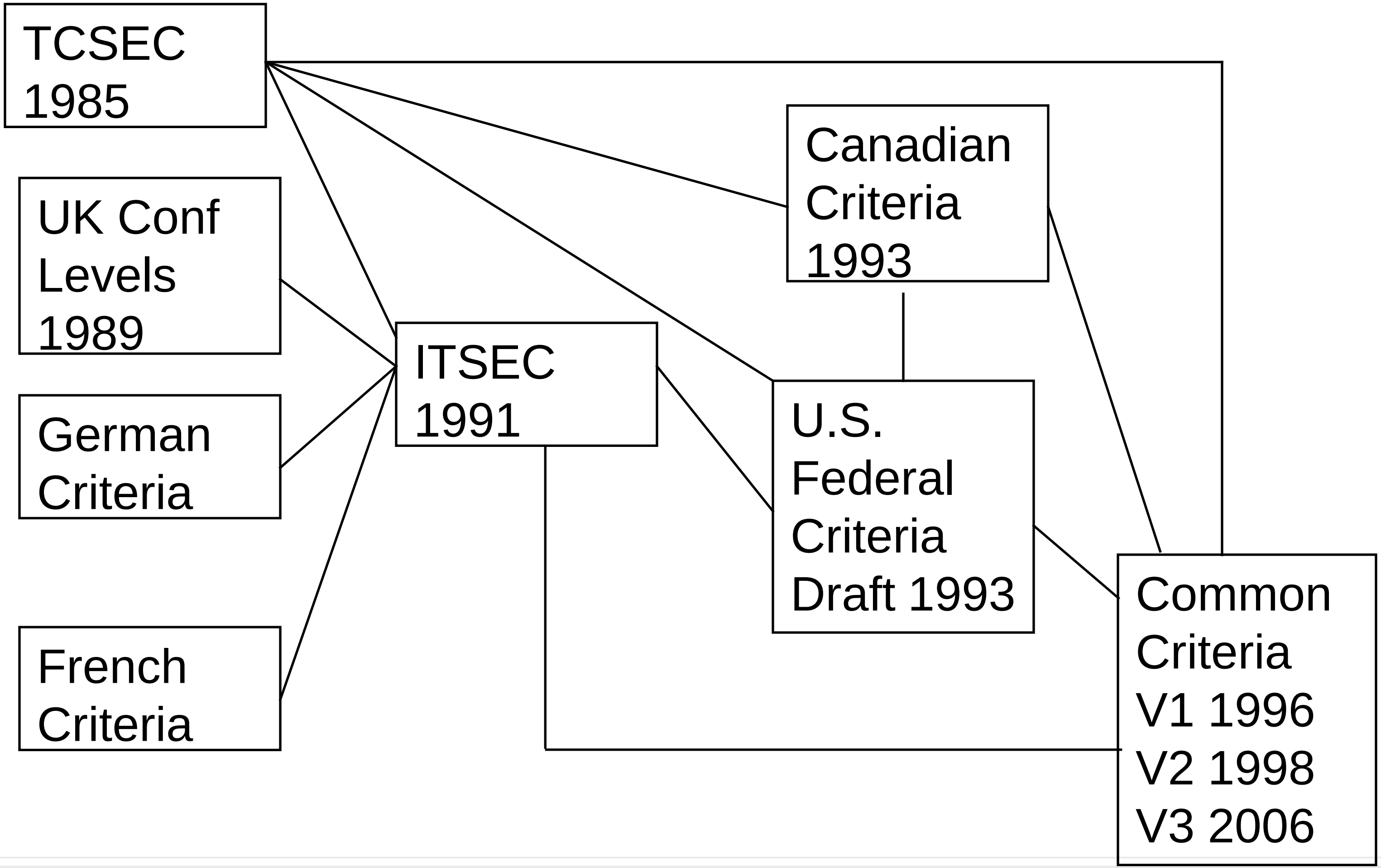
IS 15408

Common Criteria

Security Evaluation – History

- TCSEC (Orange Book), 1985: criteria for the US defense sector, predefined evaluation classes linking functionality and assurance
- ITSEC, 1990: European criteria separating functionality and assurance so that very specific targets of evaluation can be specified and commercial needs can better addressed
- Common Criteria (CC): http://www.commoncriteria.org/, http://niap.nist.gov/cc-scheme (1996)
- TCSEC and ITSEC no longer in practical use, but are commonly referred to in the literature.

Common Criteria (ISO 15408)



Target & Purpose

- Target of evaluation
  - Product: “off-the-shelf” software component to be used in a variety of applications; has to meet generic security requirements
  - System: collection of products assembled to meet the specific requirements of a given application
- Purpose of evaluation
  - Evaluation: assesses whether a product has the security properties claimed for it
  - Certification: assesses suitability of a product (system) for a given application
  - Accreditation: decide to use a certain system

Method

- Evaluations should not miss problems, different evaluations of the same product should give the same result.
- Product oriented: examine and test the product; better at finding problems.
- Process oriented: check documentation & product development process; cheaper and better for repeatable results.
- Repeatability and reproducibility often desired properties of an evaluation methodology.

Organizational Framework

- Public service: evaluation by government agency; can be slow, may be difficult to retain qualified staff.
- Private service: evaluation facilities usually accredited by a certification agency.
  - How to make sure that customer pressure does not influence evaluation results?
  - Contractual relationship between evaluation sponsor, product manufacturer, evaluation facility?
- Interpretation drift (criteria creep): meaning of criteria may change over time and differ between evaluators.

Structure

- Structure of evaluation criteria:
  - Functionality: security features
  - Effectiveness: are mechanisms adequate
  - Assurance: are mechanisms robust
- Orange Book: evaluation classes for a given set of typical DoD requirements, consider all three aspects simultaneously.
- ITSEC and CC: flexible evaluation framework that can deal with new security requirements; the three aspects are addressed independently.

Costs and Benefits

- Direct costs: fees paid for evaluation.
- Indirect costs: employee time, training evaluators in the use of specific analysis tools, impact on development process.
- When evaluating a product, the cost of evaluation may be spread over a large number of customers.
- Benefits: evaluation may be required, e.g. for government contracts; marketing argument; better security?

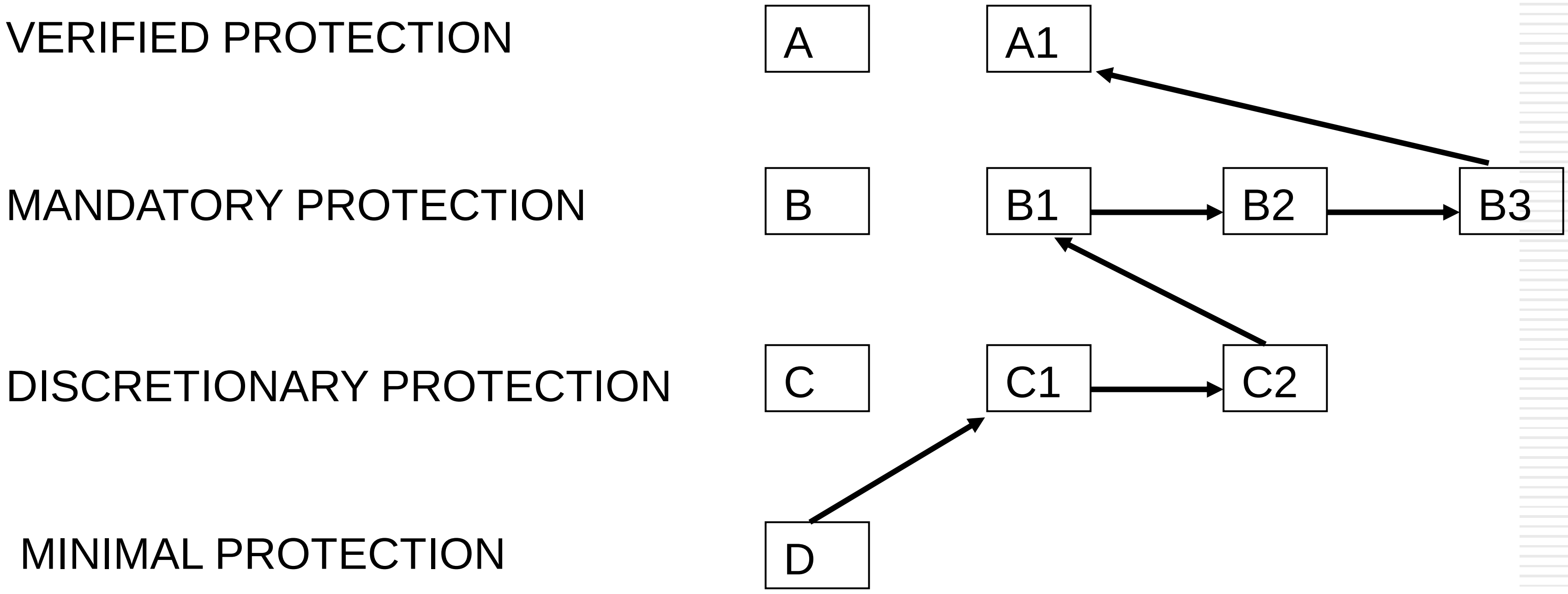
TCSEC / Orange Book

- Trusted Computer Security Evaluation Criteria
  - Published 1983. US DoD Standard 1985
  - Called Orange book because of orange cover:
- TCSEC focuses on confidentiality
- TSEC Purpose
  - Consistent set of requirements
  - Aids degree of trust in computer systems
  - Basis for specifying security requirements

TCSEC Evaluation Classes

- Four security divisions:
  - A – Verified Protection
  - B – Mandatory Protection (based on labels)
  - C – Discretionary Protection (‘need to know’)
  - D – Minimal Protection
- Security classes defined incrementally; all requirements of one class automatically included in the requirements of all higher classes.
- Class D for products submitted for evaluation that did not meet the requirements of any Orange Book class.
- Products in higher classes provide more security mechanisms and higher assurance through more rigorous analysis.

TCSEC: Evaluation Hierarchy



VERIFIED PROTECTION A A1

MANDATORY PROTECTION B B1 B2 B3

DISCRETIONARY PROTECTION C C1 C2

MINIMAL PROTECTION D

TCSEC – RAINBOW SERIES

- Orange Book – TCSEC
- Red Book – TNI (Trusted Network Interpretation)
- Blue Book – TDI (Trusted Database Interpretation)
- Yellow Book – Risk Analysis

The Common Criteria – IS15408

- Common Criteria for Information Technology Security Evaluation
- Represents the outcome of a series of efforts to develop criteria for evaluation of IT security that are broadly useful within the international community.

Common Criteria (CC) Terms

- Evaluation and Certification
- Target of Evaluation (TOE)
- Security Functional Requirements (SFRs)
- Security Assurance Requirements (SARs)
- Evaluation Assurance Level (EAL)
- Security Target (ST)
- Protection Profile (PP)

Common Criteria

- Criteria for the security evaluation of products or systems, called the Target of Evaluation (TOE).
- Protection Profile (PP): a (re-usable) set of security requirements, including an EAL; should be developed by user communities to capture typical protection requirements.
- Security Target (ST): expresses security requirements for a specific TOE, e.g. by reference to a PP; basis for any evaluation.
- Evaluation Assurance Level (EAL): define what has to be done in an evaluation; there are seven hierarchically ordered EALs.

The CC Standard

- Part 1 -Overview
- Part 2 – SFRs Security Functional Requirements
  - Security Functional Requirements (SFRs) are “what does the product does.” Taken together, the SFRs a product claims describe the product’s capabilities. A product’s security features, for example, might be how it identifies and authenticates users.
- Part 3 – SARs: Security Assurance Requirements
  - Security Assurance Requirements (SARs) define the development environment in all its phases: specification, development tools and practices, for example, the use of automated tools to prevent unauthorized modifications to the product, the completeness of test coverage.

Protection Profiles

- PPs are needed when setting the standard for a particular product type.
- PPs can be defined by government, agencies, consumers or developers.
  - PPs are published at various official websites,

    http://www.radium.ncsc.mil/tpep/library/protection_profiles/index.html
- Registration of a PP means that it is included in one or current national scheme lists

STANDARD PPs

- Organisations have produced PPs for various classes of products e.g.
  - operating systems
  - firewalls
  - smart cards
- Such PPs provides a set of functional and assurance requirements for the product in a specific threat environment

Protection Profile Examples

- Controlled Access Protection Profile (CAPP)
  - derived from TCSEC, C2 class (EAL3)
  - essentially DAC
  - NSA, October 1999
- Labelled Security Protection Profile (LSPP)
  - derived from TCSEC, B1 class (EAL3)
  - includes MAC and DAC policy
  - NSA, October 1999
- Role-based Access Control Protection Profile (RBACPP)
  - Each user has one or more roles
  - Roles may be hierarchically defined
  - CygnaCom & NIST, July 1998

CC Assurance Levels

- EAL1 - functionally tested
- EAL2 - structurally tested
- EAL3 - methodically tested and checked
- EAL4 - methodically designed, tested, and reviewed
- EAL5 - semiformally designed and tested
- EAL6 - semiformally verified design and tested
- EAL7 - formally verified design and tested

Assurance Levels

- EAL1: tester receives the target of evaluation, examines the documentation and performs some tests to confirm the documented functionality; evaluation should not require any assistance from the developer; the outlay for evaluation should be minimal.
- EAL2: developer provides test documentation and test results from a vulnerability analysis; evaluator reviews documentation and repeats some of these tests; effort required from the developer is small and a complete development record need not be available.

Assurance Levels

- EAL3: developer uses configuration management, documents security arrangements for development, and provides high-level design documentation and documentation on test coverage for review;
  - EAL3 intended for developers who already follow good development practices but do not want to implement further changes to their practices.
- EAL4: developer provides low-level design and a subset of security functions (TCB) source code for evaluation; secure delivery procedures; evaluator performs an independent vulnerability analysis.
  - Usually EAL4 is the highest level that is economically feasible for an existing product line.

Assurance Levels

- EAL5: developer provides formal model of the security policy, a semiformal high-level design, functional specification, and the full source code of the security functions; covert channel analysis; evaluator performs independent penetration testing.
  - TOE should have been designed and developed with the intent of achieving EAL5 assurance; additional evaluation costs ought not to be large.
- EAL6: source code well structured, reference monitor must have low complexity; evaluator conducts more intensive penetration testing; cost of evaluation expected to increase.

Assurance Levels

- EAL7: developer provides a formal functional specification and a high-level design, demonstrates correspondence between all representations of the security functions.
  - EAL7 typically only achieved with a TOE that has a tightly focused security functionality and is amenable to extensive formal analysis.

Certification Boards

- Operated and funded by national governments.
  - NSM (National Security Authority) in Norway
  - NIAP (National Information Assurance Partnership) (NSA & NIST) and the CCEVS (CC Evaluation and Validation Scheme) in the USA
  - GESG (Communications-Electronics Security Group) in the UK.
  - DSD (Defence Signals Directorate) in Australia

Using the Common Criteria

- CC is useful for:
  - Specifying security features in product or system
  - Assisting in the building of security features into products or systems
  - Evaluating the security features of products or systems
  - Supporting the procurement of products or systems with security features
  - Supporting marketing of evaluated products
- But
  - Evaluation is expensive and slow
  - New versions of a product must be re-evaluated, but can be done more quickly than the original evaluation.

End of lecture