INF3510 Information Security
University of Oslo
Spring 2012

Lecture 4

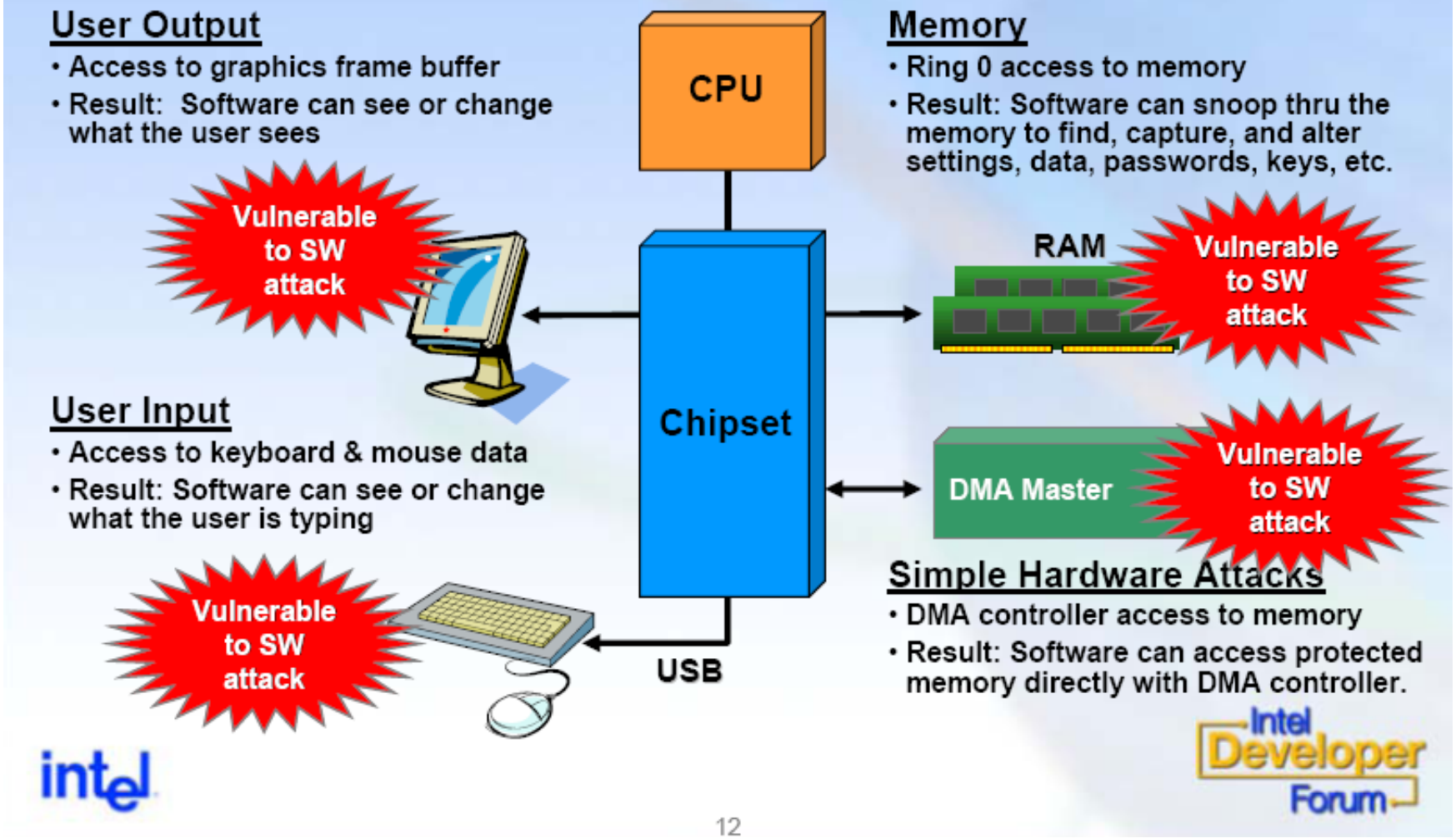
Computer Security

Audun Jøsang

Lecture Overview

- Secure computer architectures
- Virtual machines
- Trusted computing - background motivation and history
- Memory Corruption
- Security Evaluation

Vulnerabilities of the PC Today
Sample of Common Vulnerabilities



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

Meaningless transport defences when endpoints are insecure

"Using encryption on the Internet is the equivalent of arranging an armored car to deliver credit card information from someone living in a cardboard box to someone living on a park bench.“

(Gene Spafford)

Approaches to strengthening platform security

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

TCB – Trusted Computing Base

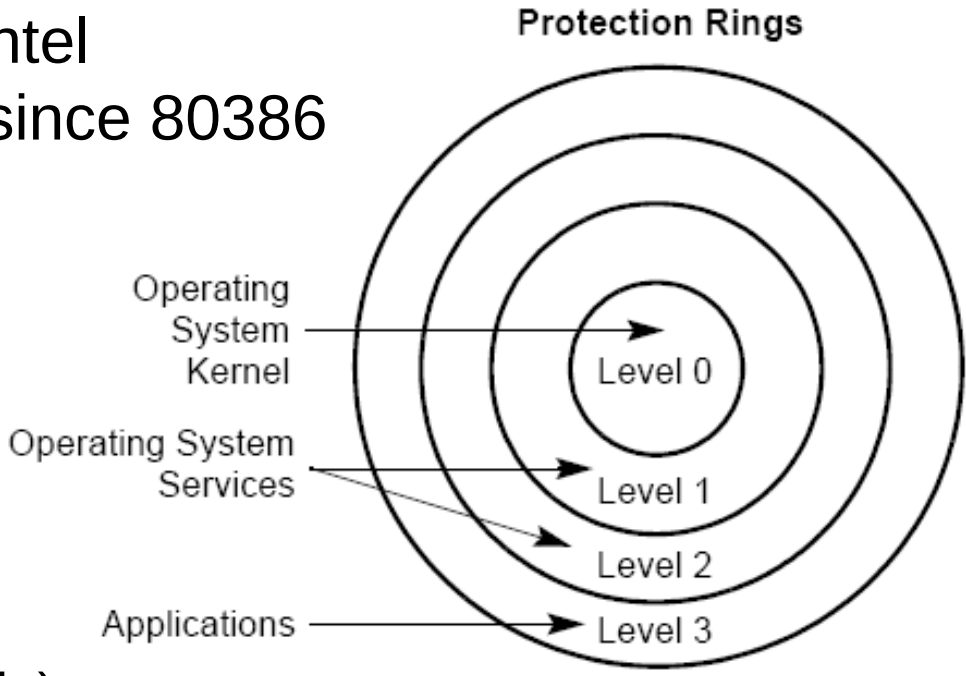
- The trusted computing base (TCB) of a computer system is the set of all hardware, firmware, and/or software components that are critical to its security, in the sense that bugs or vulnerabilities occurring inside the TCB might jeopardize the security properties of the entire system.
- By contrast, 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 a security model for enforcing 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 an example of a reference monitor placed at the lowest level.

Security kernel as reference monitor

- Hierarchic security levels in Intel microprocessor architecture since 80386
- 4 ordered privilege levels
  - Ring 0: highest
  - Ring 3: lowest
  - Intended usasge →
- Windows and Linux usage:
  - Ring 0 for OS and drivers (admin)
  - Ring 3 for applications (user space)
  - Rings 1 & 2 are not used, for performance reasons

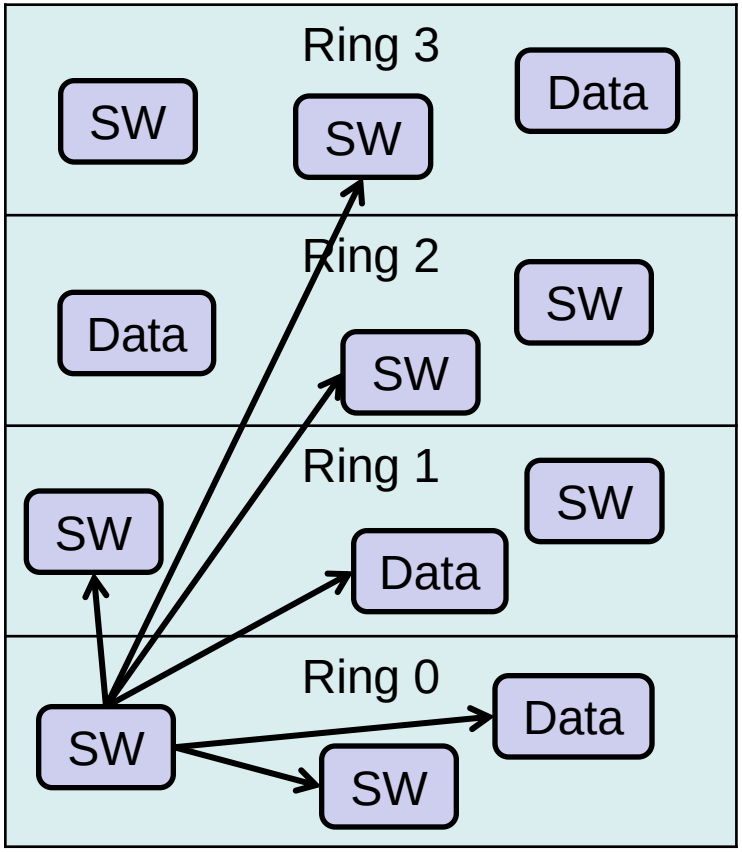


Privileged Instructions

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 attempted when the process Privilege Level is not 0, a general-protection exception (#GP) is generated, and the program crashes.

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



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

Virtual machine (VM)

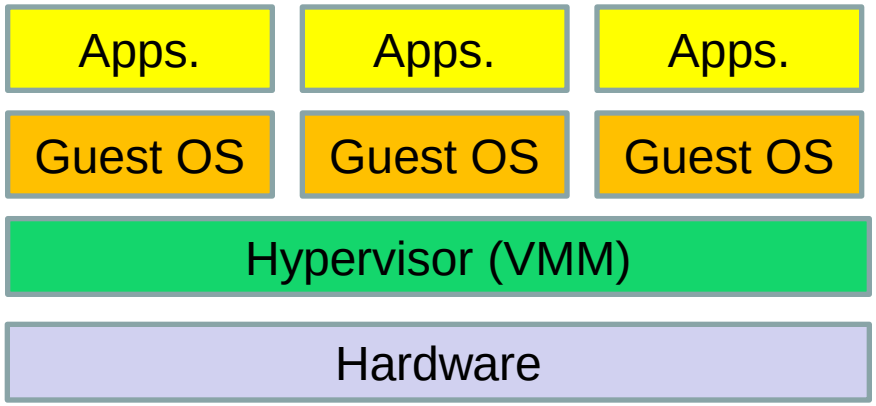
- A software implementation of a machine (computer) that executes programs like a real machine.
- Example: Java Virtual Machine (JVM)
  - JVM accepts a form of computer intermediate language commonly referred to as Jave bytecode.
    - "compile once, run anywhere“
  - The JVM translates the bytecode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

Hypervisor as reference monitor

- Platform virtualization allows multiple OSs to execute on top of a reference monitor called Hypervisor
- Each OS is a VM (Virtual Machine) controlled by the Hypervisor
- There are many Hypervisor implementations available
  - VM Ware 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

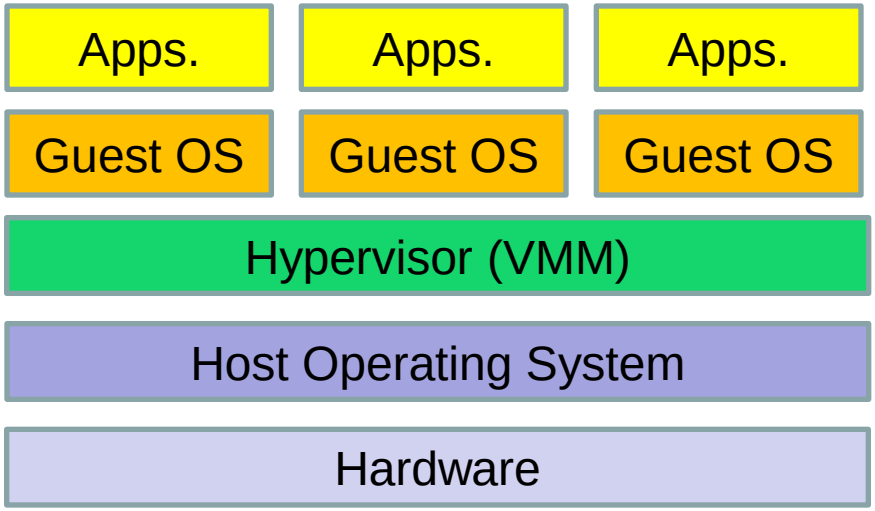
VM Architecture Variants

Type 1 VM architecture



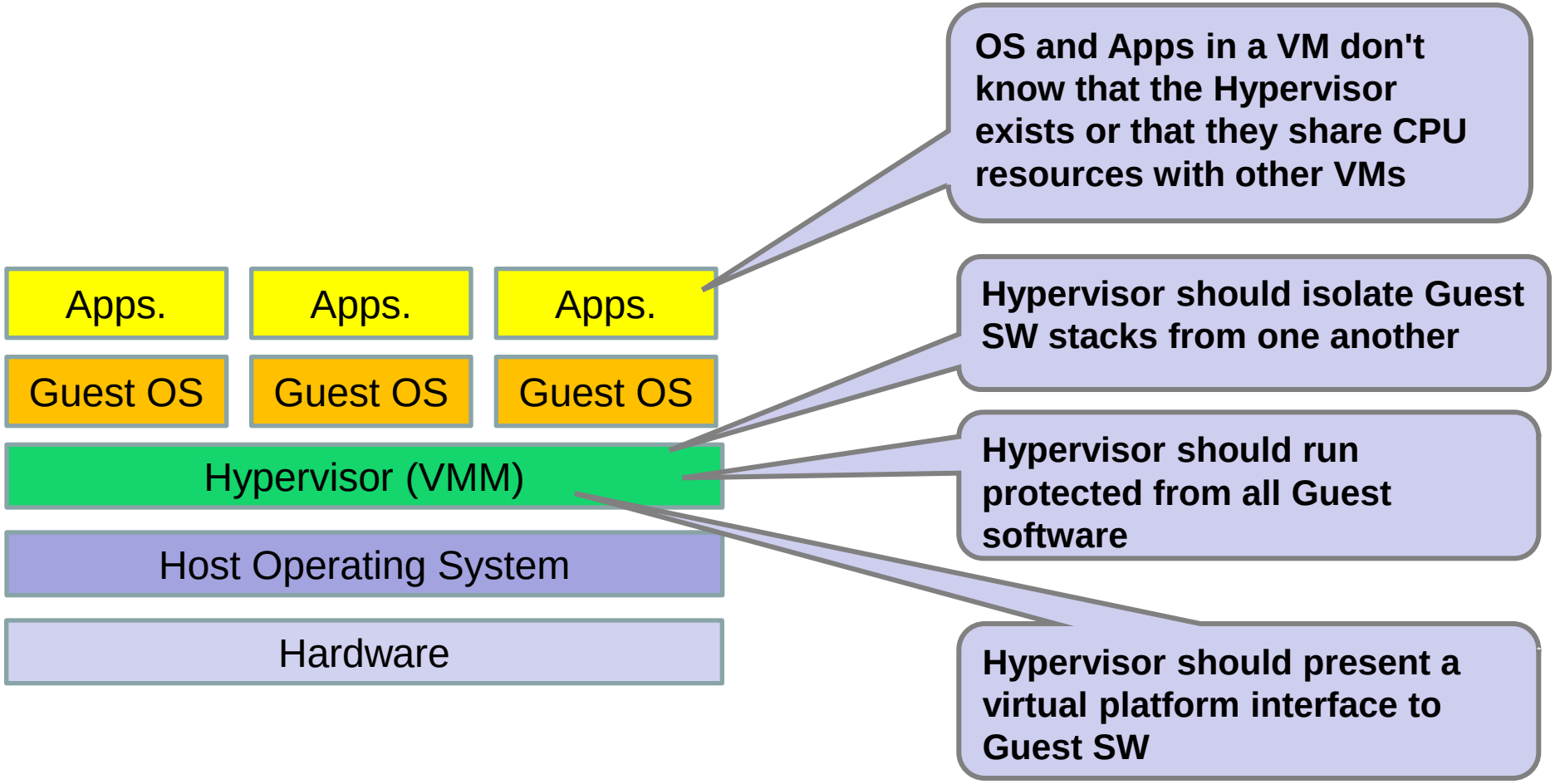
- No host OS
- Hypervisor runs on hardware
- High performance
- Limited GUI
- Suitable for servers

Type 2 VM architecture



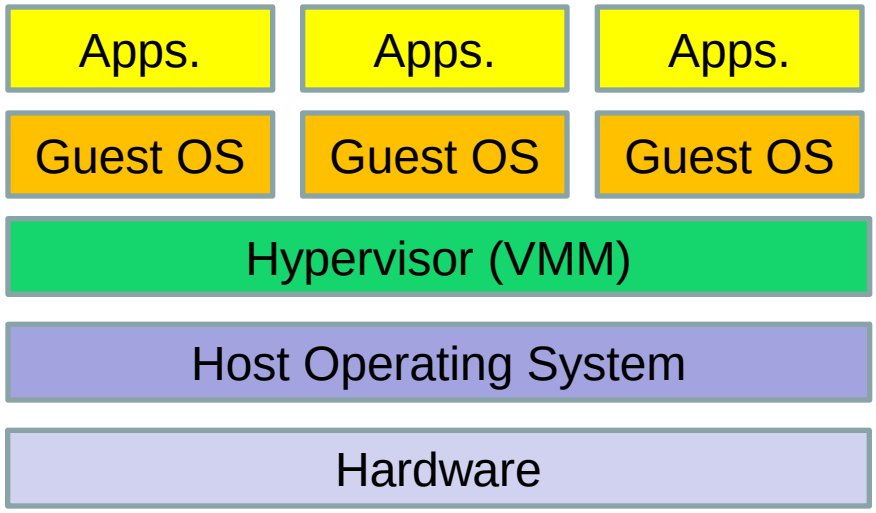
- Hypervisor runs on host OS
- Performance penalty
- Good GUI
- Better HW support
- Suitable for workstations

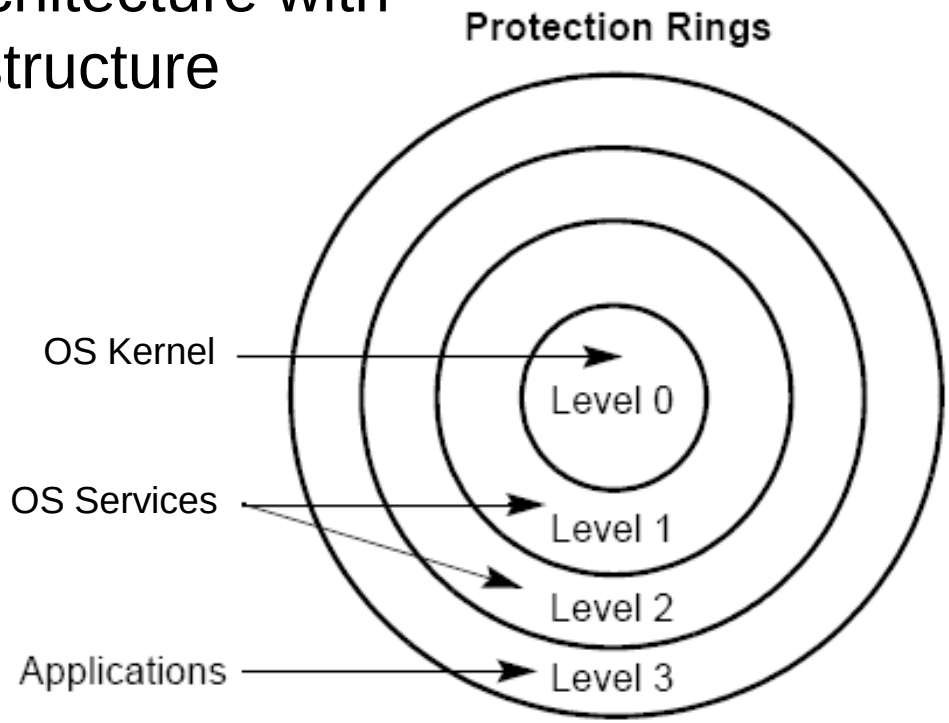
Challenges of OS Virtualisation



VM security architecture

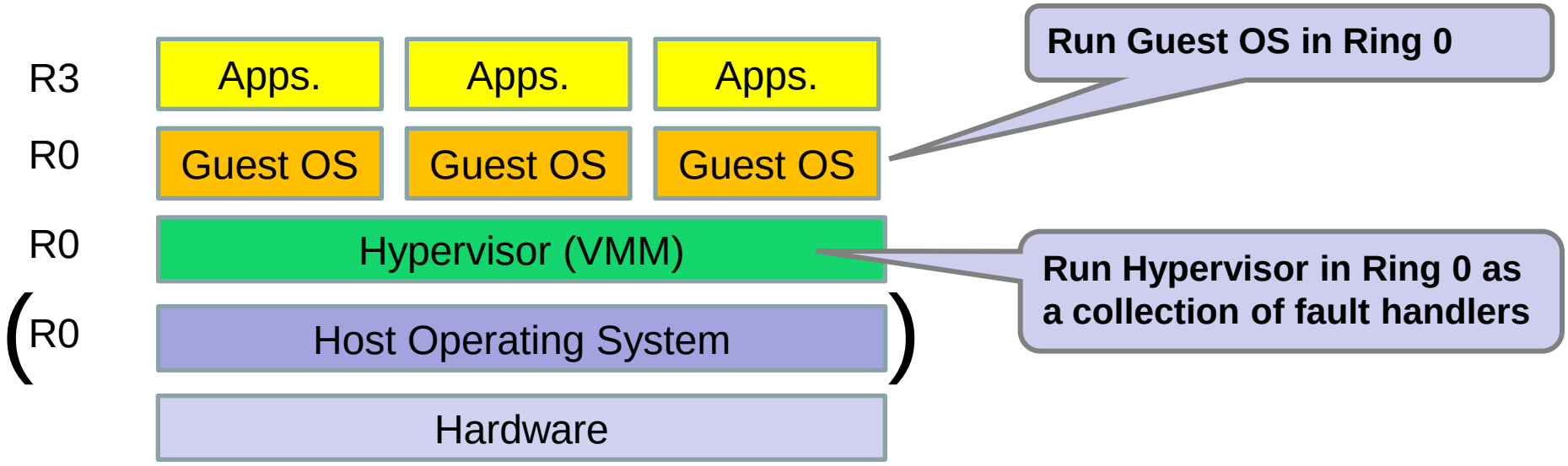
- Appropriate protection rings must be assigned to specific layers in the VM model
- Difficult to create secure architecture with traditional Protection Ring structure





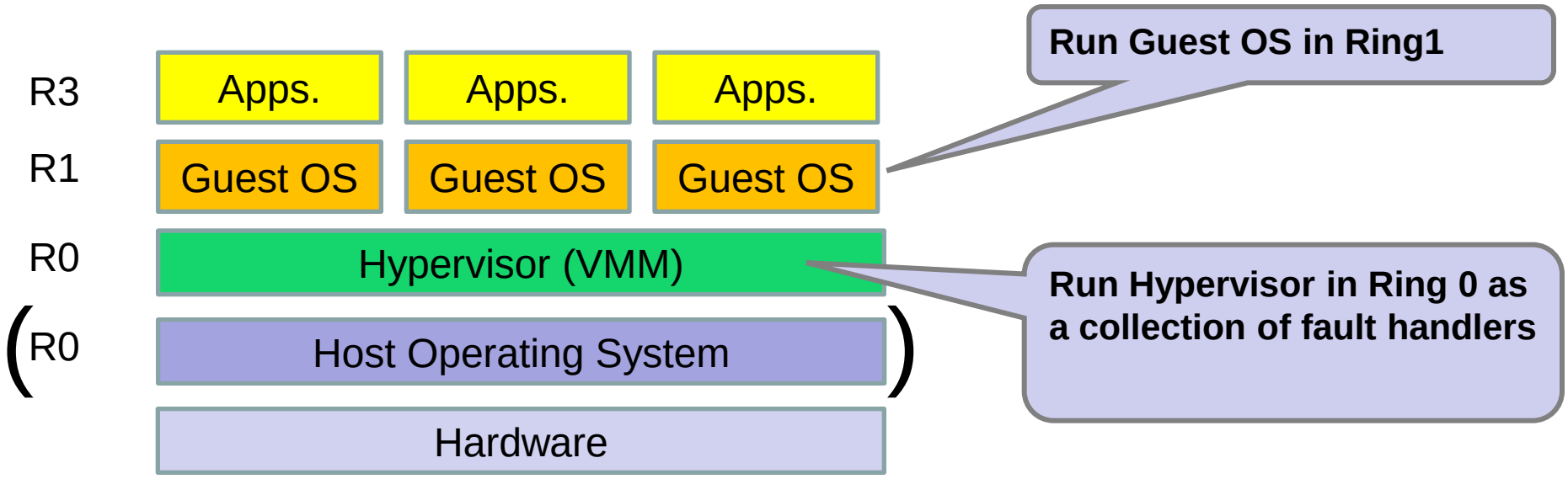
VM security architecture

Hypervisor and VMs both in Ring 0



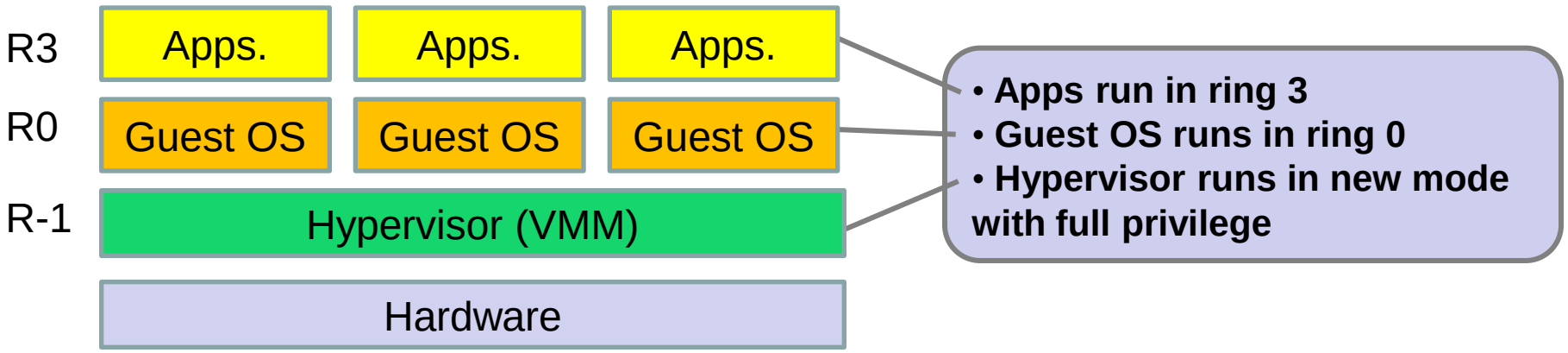
- Guest OS SW has same privilege as VMM and Host OS
- Guest OS can access VMM and Host OSS, which is bad.

VM security architecture:
VMs in Ring 1



- Virtualization of current CPUs requires complex software workarounds
- OS SW is designed to run in Ring 0
- Making an OS run in e.g. Ring 1 is tricky

VM security architecture:
Hypervisor in Ring -1



- Introducing new Ring “minus 1” eliminates virtualization holes
- Guest OS can run in Ring 0.
- Hypervisor runs in Ring -1
- No need for complex software workarounds
- Requires new hardware, available since 2005, but still not in all new systems
  - Intel Virtualization Technology (Intel VT-x)
  - AMD Virtualization (AMD-V)
- Many existing hypervisors use this model

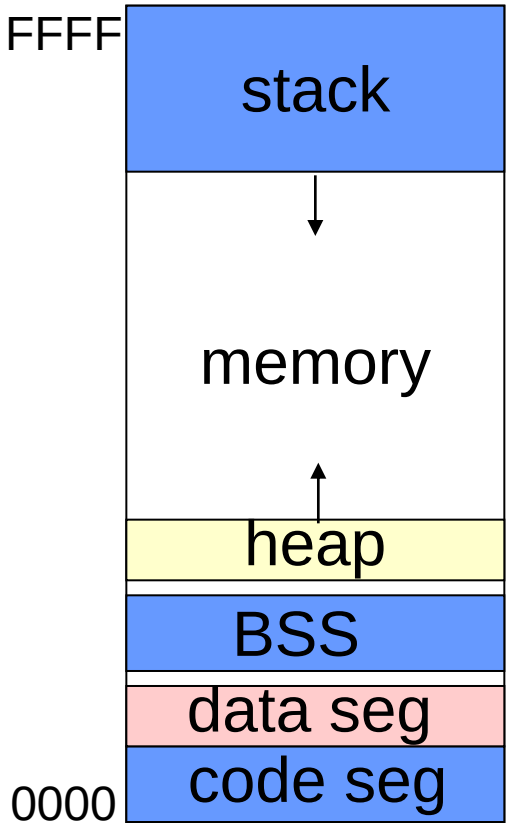
Why use a VM?

- Allows multiple OSs on same hardware
  - improved security,
  - improved management and resource utilization
  - reduced energy consumption (green hype)
- Take a snapshot of the current state of the OS
  - Use this later on to reset the system to that state
- Distribute applications bundled with OS
  - Allows optimal combination of OS and application
- Safe testing and analysis of malware
  - Malware can only infect the VM

Memory Corruption

Memory corruption and buffer overflow

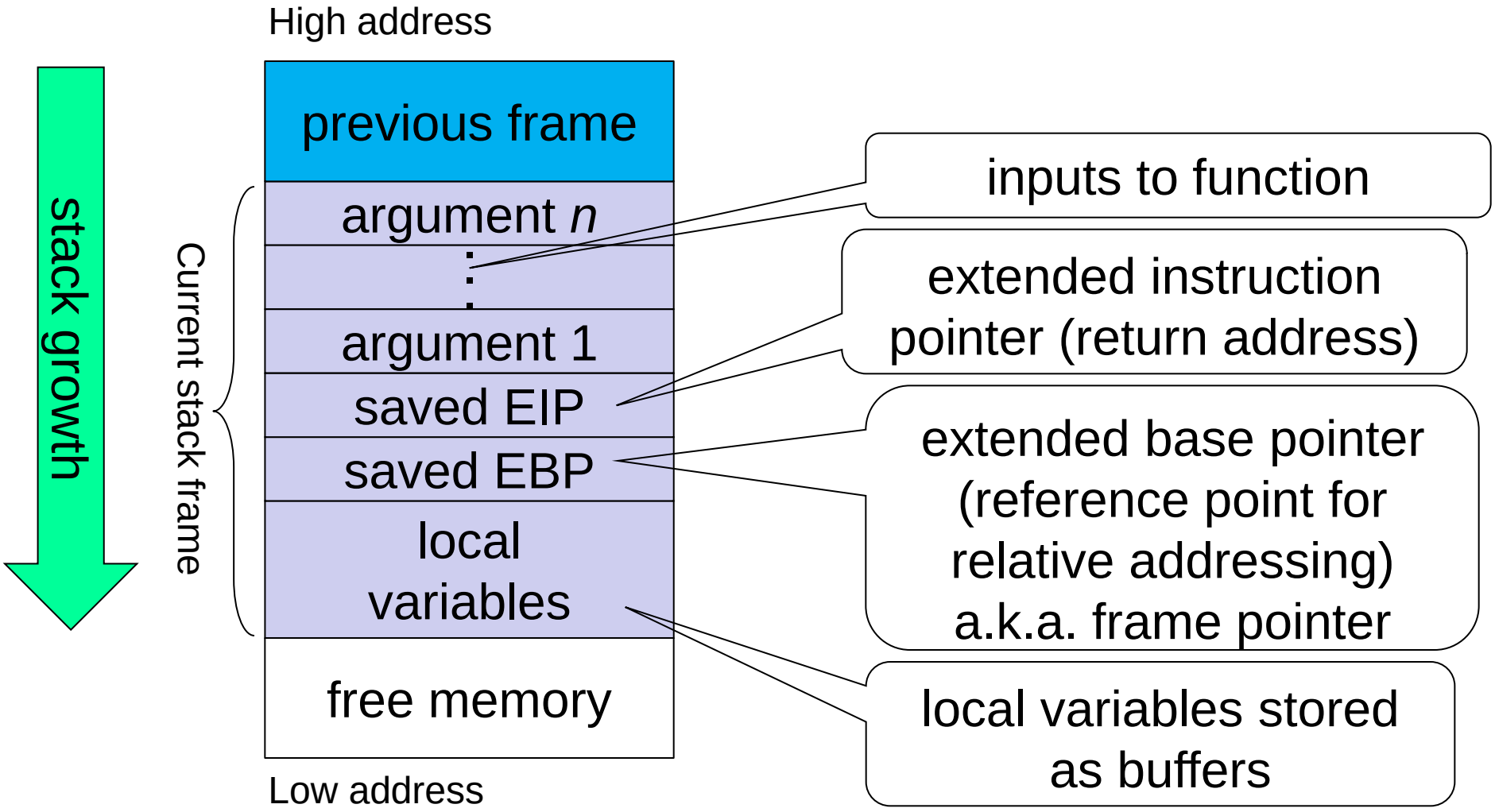
- The stack contains memory buffers that hold return address, local variables and function arguments. It is relatively easy to decide in advance where a particular buffer will be placed on the stack.
- Heap: dynamically allocated memory; more difficult but not impossible to decide in advance where a particular buffer will be placed on the heap.



Buffer Overflow

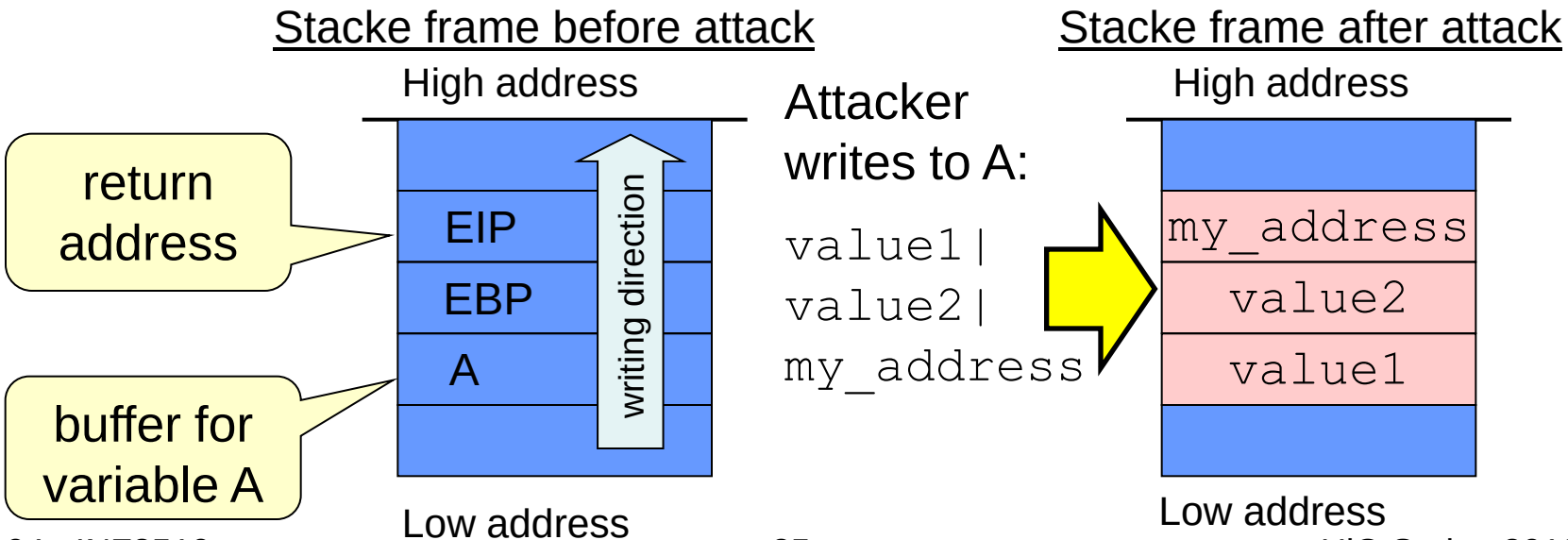
- Buffer overflow is when written data size > buffer size
  - Results in neighbouring buffers being overwritten
- Unintentional buffer overflow crashes software, and results in unreliability software.
- Intentional buffer overflow is when an attacker modifies specific data in memory to execute malware
- Targets are return addresses (specify the next piece of code to be executed) and security settings.
- In languages like C or C++ the programmer allocates and de-allocates memory.
- Type-safe languages like Java guarantee that memory management is ‘error-free’.
- Community website: http://smashthestack.org/

Stack Frame – Layout



Stack-based Overflows

- Find a buffer on the runtime stack of a privileged program that can overflow the return address.
- Overwrite the return address with the start address of the code you want to execute.
- Your code is now privileged too.



Defences against memory corruption

- Hardware functions
  - NX (No eXecute) bit/flag in stack memory
    - Injected attacker code will not execute
- OS / compiler functions
  - Stack cookies: detects corruption at runtime
  - ASLR (Address Space Layout Randomization)
    - Makes it difficult to locate functions in memory
- Programming language
  - Type safe languages like Java and C#
- Programming rules
  - Avoid vulnerable functions like
    - strcpy (use strncpy instead)
    - gets (use fgets instead)

Trusted Computing

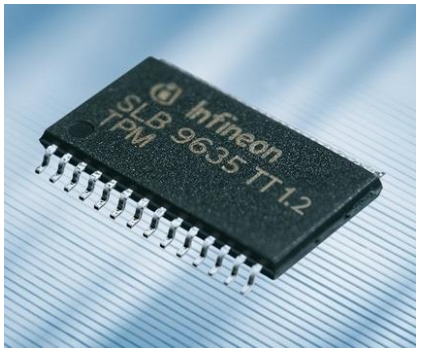
Trusted Computing: basic idea

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

- 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

Trusted Hardware Examples

TPM Chip



iButton

IBM 4764

Fortezza PC Card

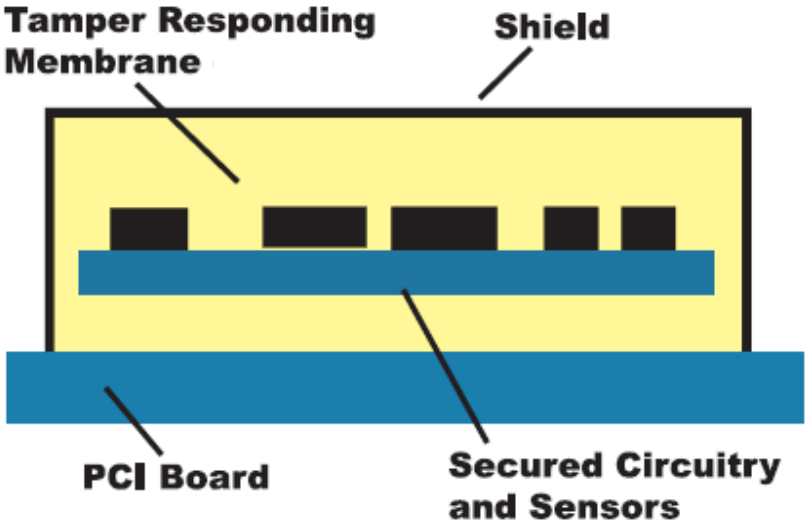
Smart Card

Characteristics of Trusted Hardware

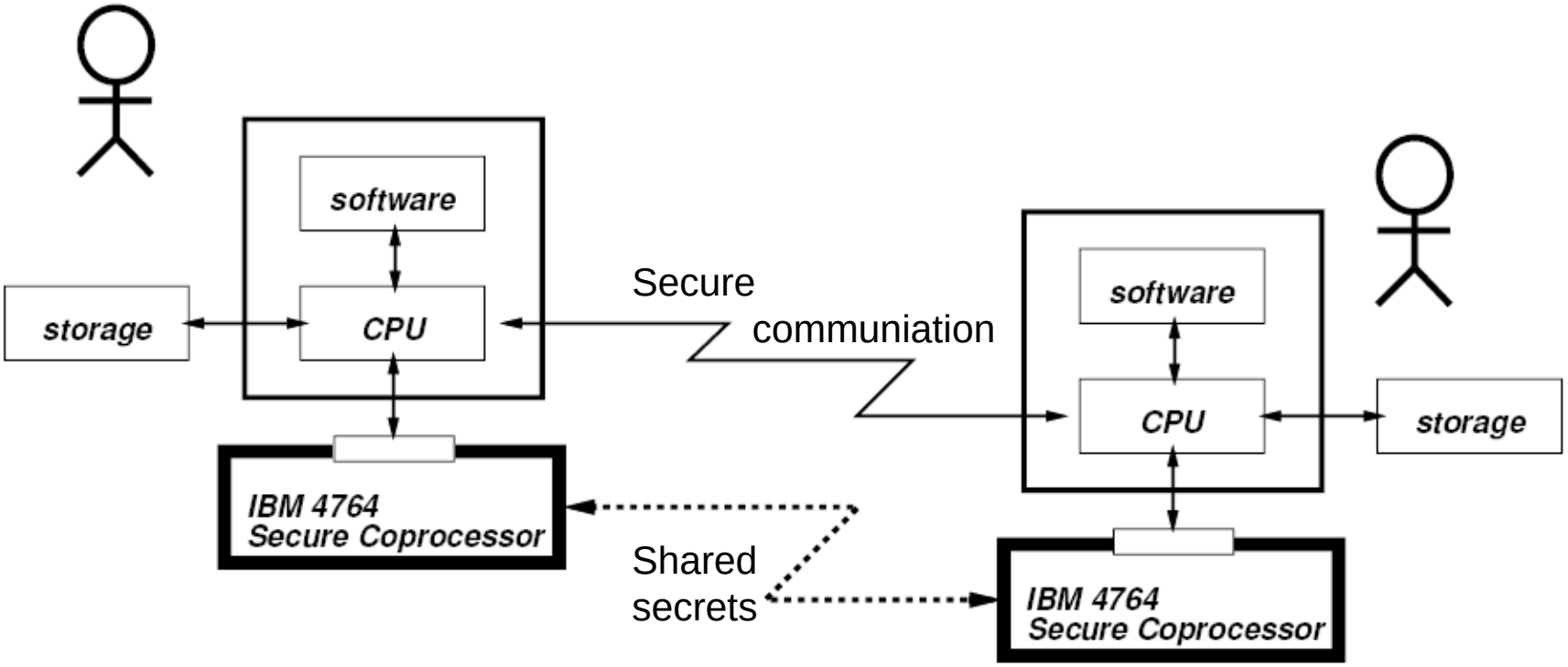
- Physically secure module
- Environmental monitoring (temperature, power supply, structural integrity)
- Tamper responsive
- Optimized hardware support for cryptography
- I/O interface

Trusted Hardware – Example

- IBM 4764 Secure Coprocessor



IBM 4764 Application Example



TCG (Trusted Computing Group)
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
  - Not-for-profit industry standards organization
- 2003: TPM specification adopted by TCG
  - Currently TPM specification 1.2
- 2010: Reduced interest
  - TPM has failed to meet industry expectations

TCG Promoters

TPM chip

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

BitLocker in Microsoft Vista / Windows 7

- 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 BitLocker Protection

BitLocker offers different types of protection, depending on needs

TPM + USB

"TPM present + what you have"

Protects Against:
HW attacks

Vulnerable To:
Stolen USB key

User Must:
Protect USB key

TPM + PIN

"TPM present + what you know"

Protects Against:
Many HW attacks

Vulnerable To:
Hardware attacks

User Must:
Enter PIN to boot

USB Only

"What you have"

Protects Against:
HW attacks

Vulnerable To:
Stolen USB key
No boot validation

User Must:
Protect USB key

TPM Only

"TPM present"

Protects Against:
Most SW attacks

Vulnerable To:
Hardware attacks

User Must:
N/A
No user impact

Ease of use

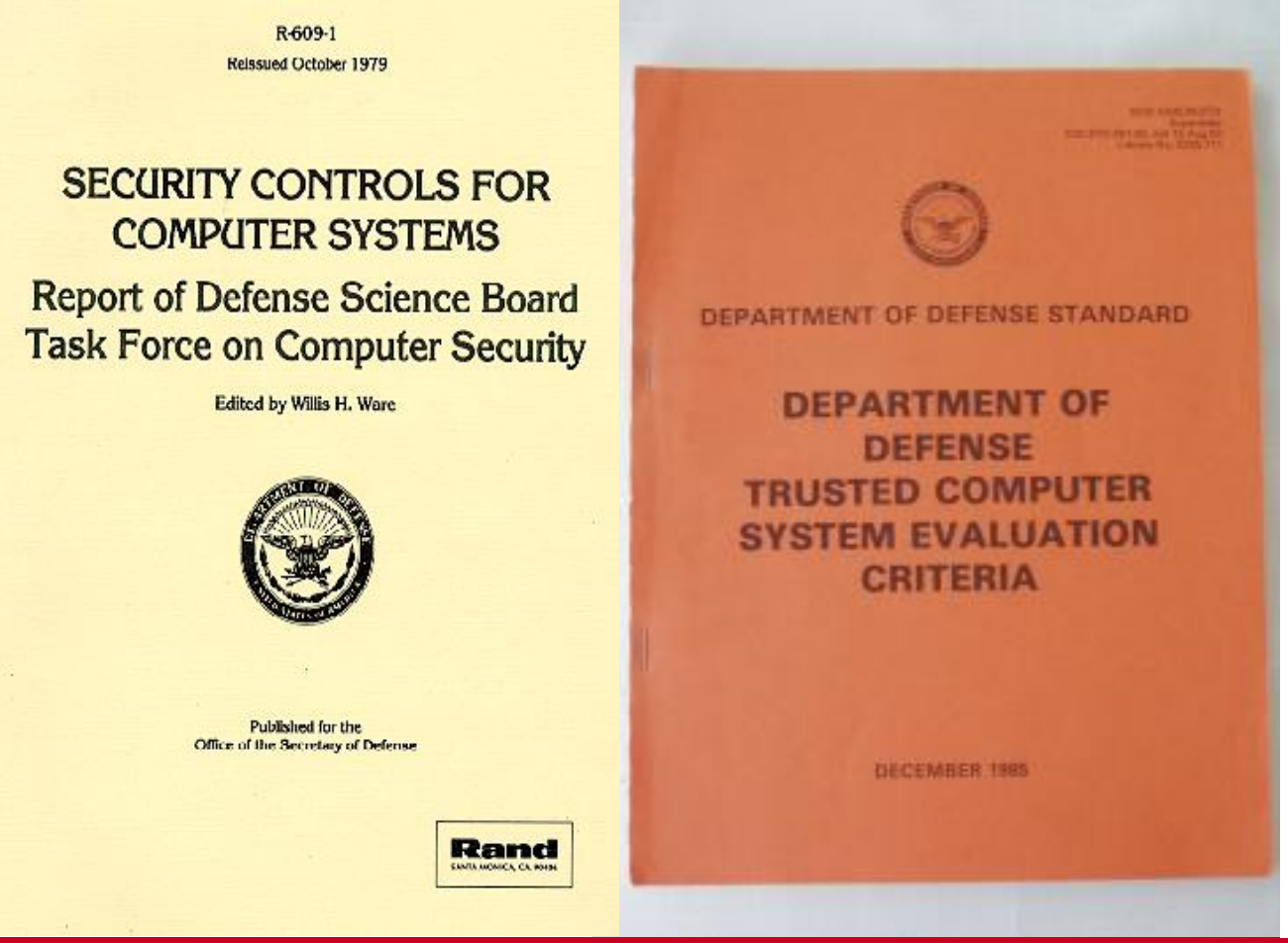
BitLocker life Cycle

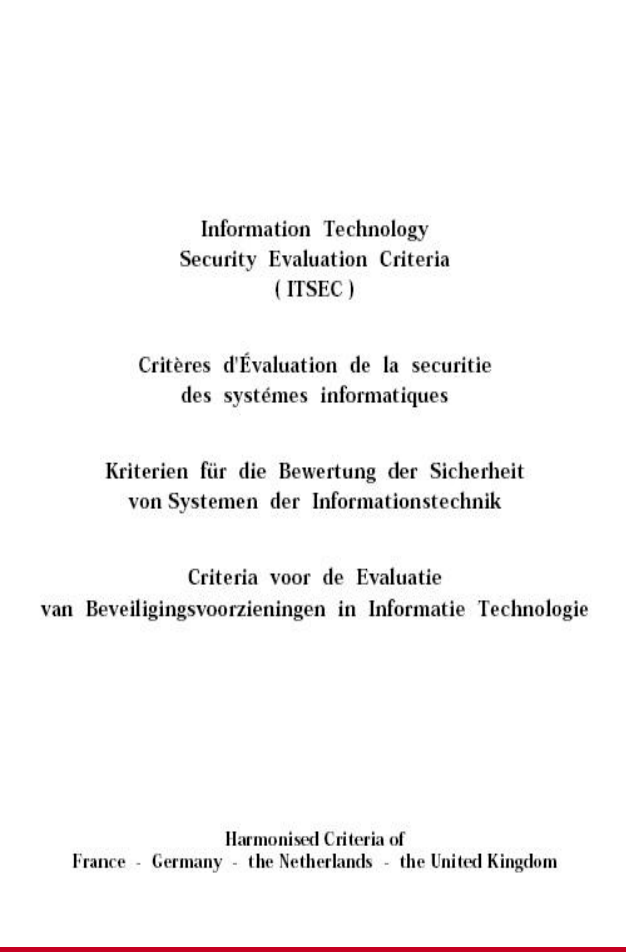
- Installation
  - Select protection
  - Select recovery password or key
- Operation - 4 different modes:
  - TPM only, TPM+PIN, TPM+USB, USB only
- Decommissioning
  - Remove keys by formatting volume
  - Remove BitLocker key protectors
  - Reset TPM

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.



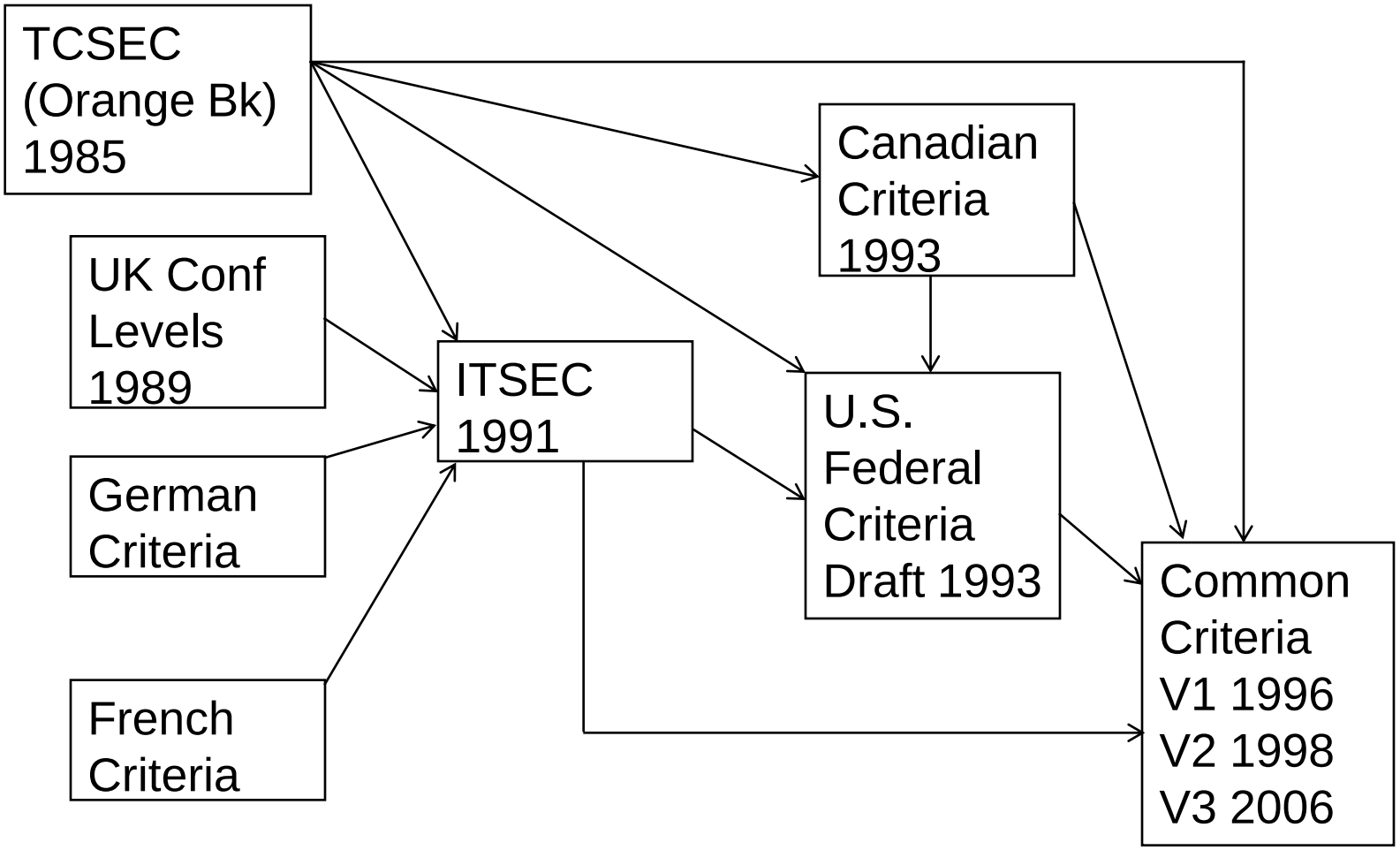


IS 15408
Common Criteria

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.

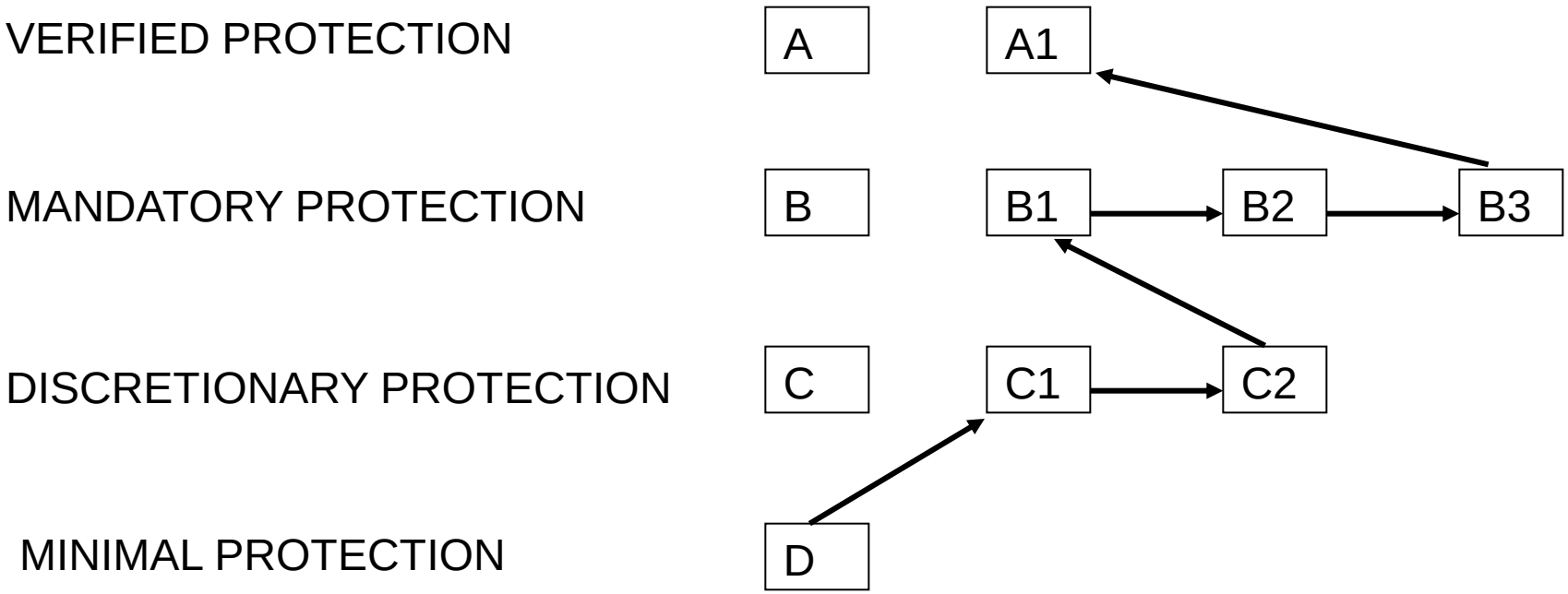
Evolution toward the Common Criteria (ISO 15408)



TCSEC Evaluation Classes

- Four security divisions:
  - A – Verified Protection (label and ACL with formal proofs)
  - B – Mandatory Protection (label + ACL based access control)
  - C – Discretionary Protection (ACL based access control)
  - 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.
- Functionality and assurance are tied together
  - Products in higher classes provide more security functionality and higher assurance at the same time.

TCSEC: Evaluation Hierarchy



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.
- Functionality and assurance aspects are independent.
  - Possible to have simple functionality with high assurance,
  - ... or to have rich functionality with moderate assurance,
  - ... or any other combination

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 the specific evaluation requirements that must be satisfied 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.
  - SERTIT/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