

Lecture 9

Computer System Security



Information & Communication Security
(WS 2010/11)

Prof. Dr. Kai Rannenber

T-Mobile Chair of Mobile Business & Multilateral Security
Goethe University Frankfurt a. M.

- Introduction
- Security Threats
- Operating System Security
- Improving Security

2008 landmark judgement by German Bundesverfassungsgericht (BVerfG):

Basic right to confidentiality and integrity of IT systems

“1. Das allgemeine Persönlichkeitsrecht (Art. 2 Abs. 1 i.V.m. Art. 1 Abs. 1 GG) umfasst das Grundrecht auf Gewährleistung der Vertraulichkeit und Integrität informationstechnischer Systeme.“

Stealthy infiltration of IT systems is problematic:

„2. Die heimliche Infiltration eines informationstechnischen Systems, mittels derer die Nutzung des Systems überwacht und seine Speichermedien ausgelesen werden können, ist verfassungsrechtlich nur zulässig, wenn tatsächliche Anhaltspunkte einer konkreten Gefahr für ein überragend wichtiges Rechtsgut bestehen. (...)”

- Introduction
- Security Threats
 - Malicious Logic
 - Buffer Overflow
 - Mobile Code
- Operating System Security
- Improving Security

- Introduction
- Security Threats
 - Malicious Logic
 - Buffer Overflow
 - Mobile Code
- Operating System Security
- Improving Security

Definitions

- Is a set of instructions that cause a site's security policy to be violated.

[M. Bishop, Introduction to Computer Security]

- A program implemented in hardware, firmware, or software, and whose purpose is to perform some unauthorized or harmful action.

[ISO/IEC 2382-8]

- Hardware, software, or firmware capable of performing an unauthorized function on an information system.

[National Information Systems Security (INFOSEC) Glossary 2000]

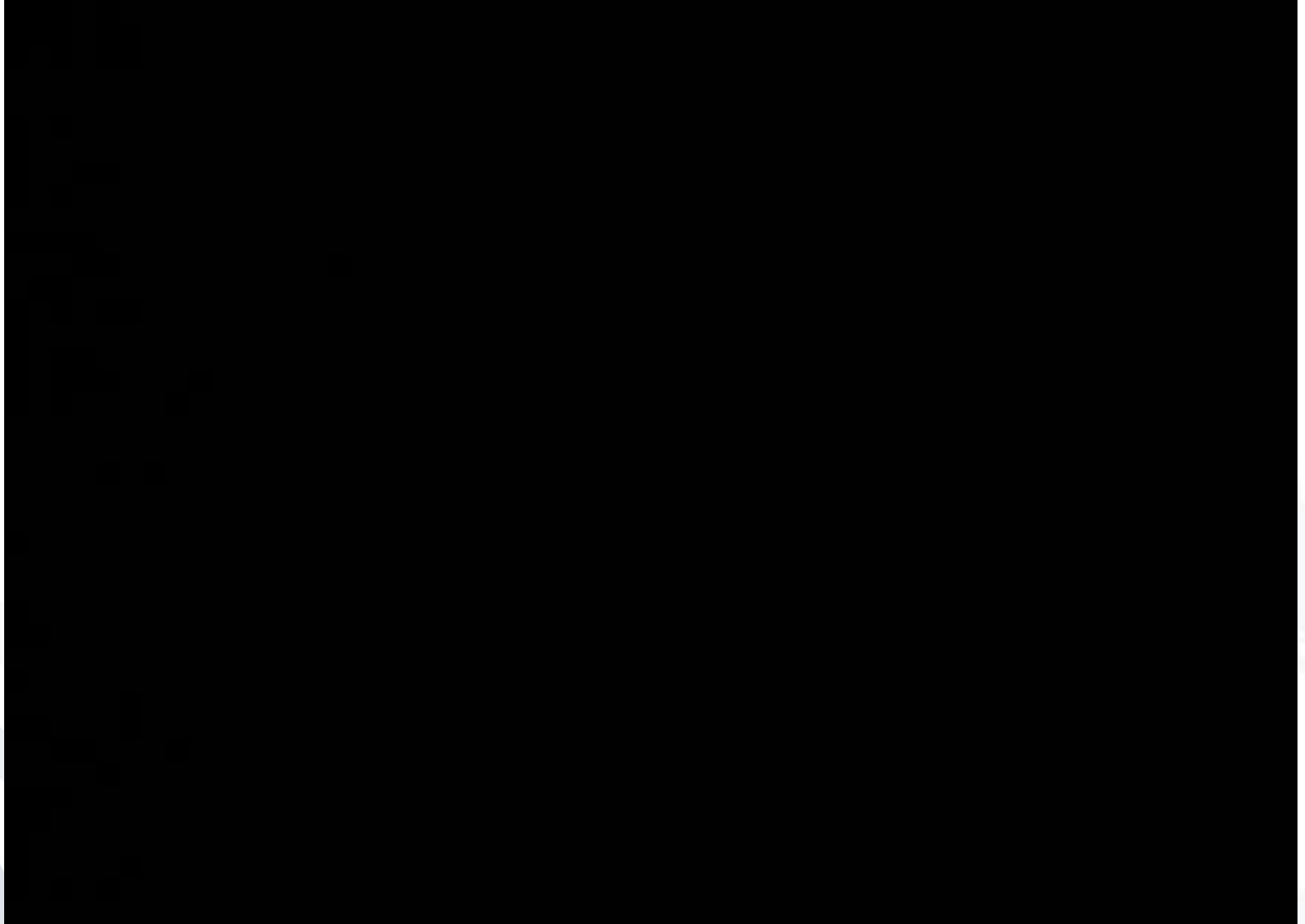
Malicious logic is also known as malicious code or **malware** (Malicious software).

- Trojan Horses
 - Programs with a covert purpose
- Viruses
 - Programs that replicate
- Worms
 - Propagate autonomously from system to system
- Logic Bombs
 - Hidden code, triggered by external event



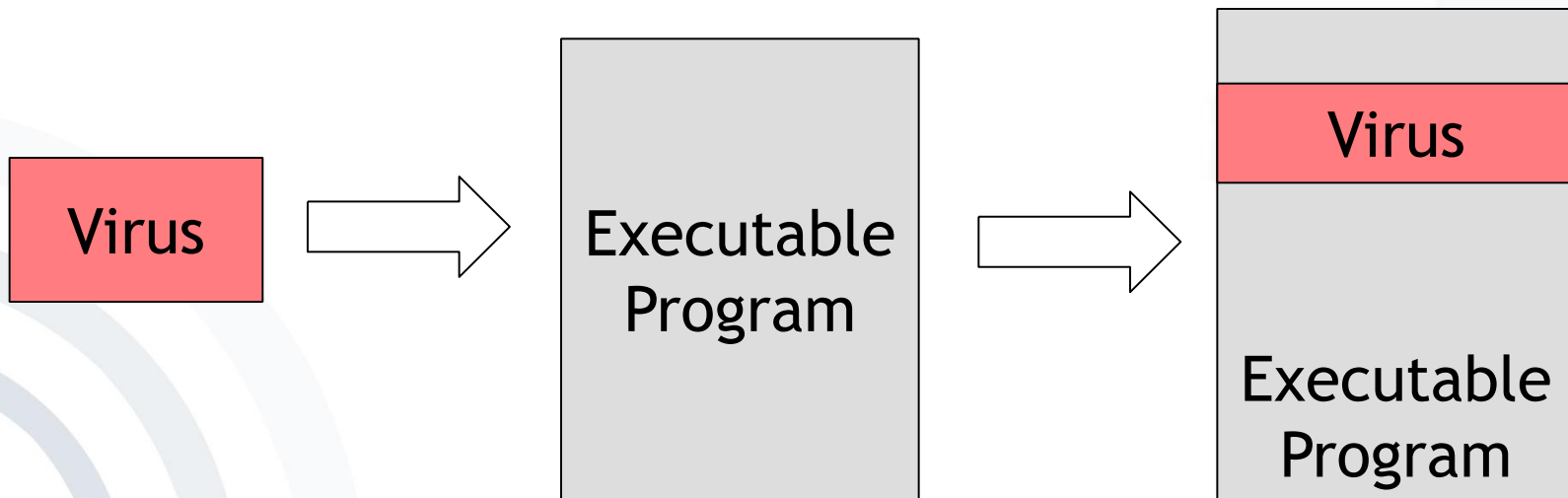
- Program with an *overt* purpose (known to user) and a *covert* purpose (unknown to user)
 - Often called a Trojan
 - Named by Dan Edwards in Anderson Report [Anderson72]
- Example: NetBus
 - Designed for Windows NT system
 - Victim uploads and installs it:
 - Usually disguised as a game program, or integrated within one
 - Acts as a server, accepting and executing commands for remote administrator
 - This includes intercepting keystrokes and mouse motions and sending them to attacker.
 - Also allows attacker to upload, download files

Example Trojan Horses



Program that replicates itself, e.g. by inserting itself into one or more files, and that may perform some other action, too:

- *Insertion phase*: Virus is inserting itself into a file.
- *Execution phase*: Virus is performing some (possibly null) action.



- Boot Sector Infector
 - Inserts itself into the boot sector of a disk
- Executable Infector
 - Infects executable programs, e.g. .EXE or .COM programs
 - May prepend itself (as shown) or put itself anywhere, fixing up binary so it is executed at some point
- Multipartite Virus
 - Can infect multiple platforms (e.g. either boot sectors or executables)
- TSR Virus (Terminate and Stay Resident)
 - Stays active in memory after the application is completed
- Stealth Virus
 - Conceals its presence on a system

- **Encrypted Virus**
 - Is enciphered except for a small deciphering routine
- **Polymorphic Virus**
 - Changes its form each time it inserts itself into another program
- **Macro Virus**
 - Composed of a sequence of instructions that are interpreted rather than executed directly
 - Can infect either executables (Duff's shell virus) or data files (Highland's Lotus 1-2-3 spreadsheet virus)
 - Independent of machine architecture
- **Retro Virus**
 - Attacks anti-virus software present on the system

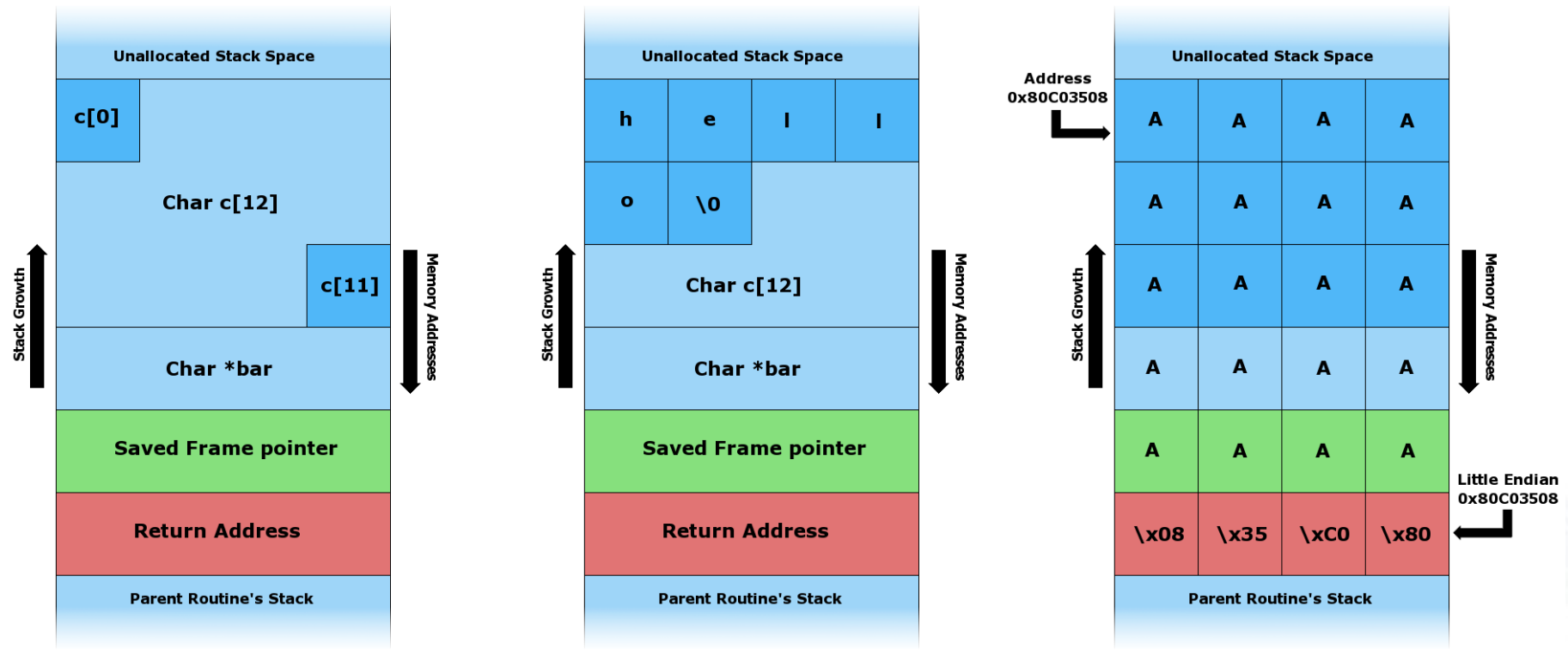
- A program that copies itself from one computer to another
- Origins: distributed computations
 - Animations, broadcast messages
- Segment
 - part of program copied onto workstation
 - processes data, communicates with worm's controller
 - Any activity on workstation causes segment to shut down.

- A program that performs an action that violates the site security policy when some external event occurs
- Example: program that deletes company's payroll records when one particular record is deleted
 - The “particular record” is usually that of the person writing the logic bomb.
 - If/when the person is fired and the payroll record deleted, the company loses *all* those records.

- Introduction
- Security Threats
 - Malicious Logic
 - Buffer Overflow
 - Mobile Code
- Operating System Security
- Improving Security

- Buffer overflow software bug
 - data larger than the variable allocated for it
 - can overwrite a procedure return address in the procedure call stack in memory
- Persisted for decades
 - Users do not bother to install patches supplied (free) by software vendors.
- Example of vulnerability that permits remote injection of hostile code, recruiting bot nets for later DDoS attacks

Buffer Overflow (2)



Before data is copied.

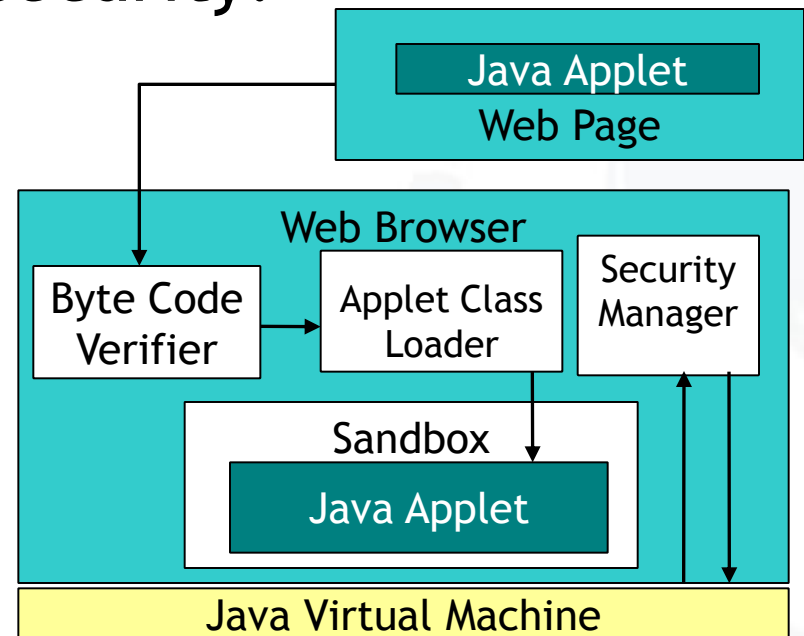
"hello" is copied.

"AAAAAAAAAAAAAAAAAAAAAA
AA\x08\x35\xC0\x80" is
copied.

- Introduction
- Security Threats
 - Malicious Logic
 - Buffer Overflow
 - Mobile Code
- Operating System Security
- Improving Security

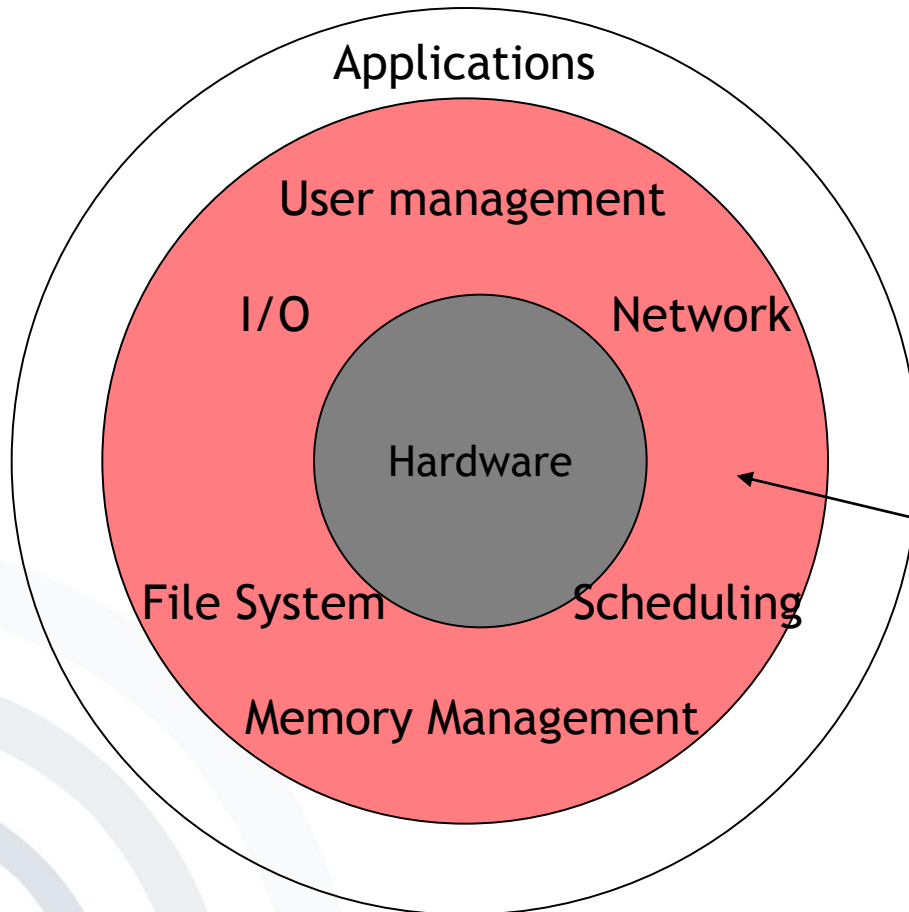
- Download code and run it locally
- Two common mobile code systems:
 - Java Virtual Machine
 - Microsoft ActiveX, .NET
- Java Virtual Machine security:

- Environment maintains a security manager that cannot be replaced.
- Security managers can be set to prevent access to local resources.
- Downloaded classes are stored separately from local classes.
- Byte code is checked for validity.



- Introduction
- Security Threats
- Operating System Security
 - Unix
 - Windows
 - Mobile OSs
- Improving Security

OS Security Considerations



- The whole code which is executed in the kernel is very security critical (supervisor mode!)
- It is the target of a lot of attacks to gain root privileges by manipulating kernel functions.

OS Kernel

- Identification
 - Recognition of human individuals
- Authentication
 - Secure confirmation of users' identifiers
- Access Control
 - Restricting usage of a service to authorized users
- Audit
 - Monitoring of system activities

- Introduction
- Security Threats
- Operating System Security
 - Unix
 - Windows
 - Mobile OSs
- Improving Security

- In the Unix operating system there are two parts:
 - Kernel
 - User space
- Any programming code in the kernel space has full access to the computer it is running on.
- Code running in the user space has access rights based on the User ID (UID) it is running under:
 - UID 0 is reserved for the super user or root and the kernel automatically gives this UID complete access.
- Note the difference between kernel and root access:
 - Kernel processes can access anything.
 - Root processes can order the kernel to access anything.

- IDs
 - User Identification number (UID)
 - Root/super user (UID 0)
 - Group Identification number (GID)
- Authentication
 - Password: /etc/passwd
 - User name (login name)
 - Password, encrypted
 - usually modified DES (or MD5, SHA...)
 - One way function, it is impossible to decrypt the password.
 - At login the entered password is encrypted and compared to file.
 - User id (number)
 - Login group id (number)
 - GCOS (Comment, usually real-life name)
 - Home directory
 - Program to be executed at login, usually shell

- Access Control
 - A file has owner and group id (sometimes several).
 - A process has owner and group id (sometimes several).
 - Kernel verifies permissions before executing system calls:
 - If owner uid=0 (root), everything is allowed
 - Otherwise the uid and gid of the process and object are compared in this order and permission for the operation is searched for based on owner, group and other (world) rights
- Auditing
 - Permanent Logging
 - Automatically recorded events
 - Manually set logging

- Introduction
- Security Threats
- Operating System Security
 - Unix
 - Windows
 - Mobile OSs
- Improving Security

- Provides security controls access and auditing
- Implements the standard subject/object security model
 - Subject - process or thread running on behalf of the system or an authenticated user
 - Object - individually secured entity such as a file, pipe, or even a process. Access control may vary between different objects.
 - Kernel mode, User mode
- Controls applied to core OS elements like processes and sockets in addition to the more traditional file system elements (NTFS).
- Problems
 - Unexpected use of extensible elements like word macros or extensible DLL's
 - Unprotected file systems
 - Attempts at backwards compatibility with older version of Windows caused some security problems (NetBIOS and FAT).

- Identification
 - User Account
 - Security ID (SID) - A globally unique ID that refers to the subject (user or group)
- Authentication
 - Password, stored as hash value
 - Secure attention sequence CTRL+ALT+DEL
 - Security Accounts Manager
- Access Control
 - Object - Individually secured entity such as a file, pipe, or even a process
 - Rights - actions associated between object and subject (Read, write, execute, audit)
 - Access token - the runtime credentials of the subject

- Access Control
 - Access control list (ACL)
 - Associated with an object
 - Ordered list
 - Each access control entry (ACE) contains a subject and a right.
 - Evaluated by the security subsystem to determine access to protected objects
 - Discretionary ACLs control access
 - System ACLs control audit
- Auditing
 - Security Reference Monitor
 - Local Security Authority
 - Event Logger
 - If auditing applies and what is to be audited is determined by the Audit Policy

- Introduction
- Security Threats
- Operating System Security
 - Unix
 - Windows
 - Mobile OSs
 - PocketPC
 - Palm OS
 - Symbian
 - Linux
- Improving Security

Once upon a time...

- Closed platforms
- No additional software could be installed.
- Limited functionality



- Open platforms
- Lots of software can be installed:
 - For different purposes
 - From different vendors
- Communication with different protocols possible:
 - GSM/GPRS, UMTS
 - Bluetooth, Infrared, WLAN
- Private and confidential data can and will be stored on the mobile device.
- Camera is (in many cases) included.



- Risks of Malware
 - Viruses, Worms, Dialler, Trojan Horses, etc.
- Passwords can (and will most likely) be deactivated.
- External storage media enables potential attackers to steal private information.
- Different communication protocols can be used to attack device or steal data.
- Camera also introduces new risks:
 - Stealing paper based confidential information
 - Invasion of personal privacy



- Offers no encryption of data
- No memory protection
- Supports ActiveX and Java
- Passwords deactivated by default
- Each application can adjust its priorities, terminate other applications or can access their memory.
- External storage media enables potential attackers to steal private information.
- No protection from malware
- Code signing mechanisms exist, but:
 - Only origin of software can be checked.
 - Administration of certificates is not possible
- Additional security software exists, but cannot resolve design failures.



- No effective distribution of rights and separation of processes
- No secure path between applications and kernel
- Can be synchronized with any PC
- Unsatisfactory password protection
- No certificate management
- Manipulated program could act with all user authorization.
- Additional security software exists, but there are still security risks.



- Provides better protection than Palm OS and PocketPC 2003
- Device can be administered by ACLs.
- Certain contents can only be synchronized with certain servers.
- Certificate management is installed, but the user cannot check the security status of the device.
- Hardly any additional security software is available.



- The unit of trust is the OS process:
 - Trusted Computing Base: the kernel, file server and, on open devices, the software installer and its registry
 - Trusted Computing Environment: other system components
 - Applications
- Capabilities are used to control access to sensitive resources:
 - Every process has a set of capabilities, and its capabilities never change during its lifetime.
 - The capabilities are stored in the kernel's memory.
 - Capabilities must be checked for each use of service.
- Data caging protects files against unauthorised access:
 - File access control

- Provides user with largest number of security functions
- Security gaps exist, while not being as prominent.
- 2005:
 - *“We think that 2005 is going to be a real breakout year for Linux on cellphones,” Troltech CEO Haavard Nord told LinuxDevices.com. “There seems to be a huge interest in Linux. Linux is just now getting mature for the market. Currently, we’re working with more than 20 manufacturers [who are] building Linux phones today,” he said. [LinuxDevices.com]*
- 2010:
 - Nearly every manufacturer
 - E.g. Samsung i7500
 - For details see:
 - www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Linux-Mobile-Phones/



Security of Operating Systems

Operating System	Memory Protection	File Protection	Access Control	Support for Security Modules	Secure I/O	Code Integrity Mgt.
Symbian 9.x	✓	✓	✓	(✓)	x	✓
PalmOS 5.x	x	x	x	(✓)	x	x
Windows CE 6.2 Windows Mobile 6.5	✓	✓	✓	(✓)	x	✓
PocketPC 2003 Phone Edition	x	x	✓	(✓)	x	x
Embedded Linux	✓	(✓)	✓	(✓)	x	x
Android	✓	✓	✓	(✓)	x	x
J2ME	✓	x	x	(✓)	x	✓

(✓) → Feature is available, depending on the available hardware (e.g. availability of a card reader).

- Introduction
- Security Threats
- Operating System Security
- Improving Security
 - Security Enhancing Techniques
 - Trusted Computing
 - TCG
 - TCG based Initiatives
 - Mobile Trusted Computing

- **Virus scanners** try to identify viruses according to a certain characteristic (virus signature) stored in a database.
- **Code Signing** helps to distinguish authorized code from other code.
- **A Trusted Operating Base** can then prohibit the execution of not authorized code e.g. viruses on a system.
- **Checksums and/or Encryption** make it possible to detect/avoid modifications done by a virus.
- **Intrusion Detection Systems (IDS)** monitor a system to detect processes which may be the result of a virus infection.
- **Heuristic virus scanners** try to identify a virus with a forecast about the runtime behaviour of code (sophisticated approach, but not really efficient).

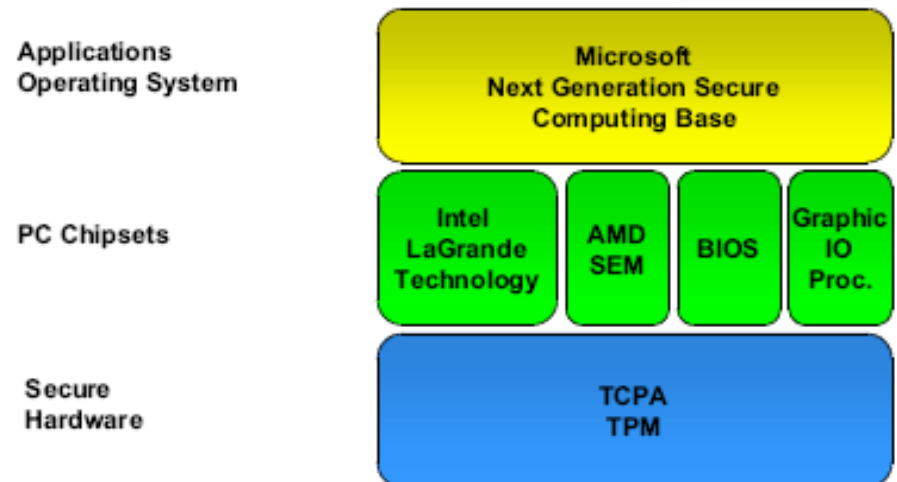
- Trust

“An entity can be trusted if it always behaves in the expected manner for the intended purpose.” [*Sandhu, Zhang 2005*]
- Entity
 - a platform, or an application or service running on a platform
 - A platform can be a personal computer, personal digital assistant (PDA), smart phone, etc.
- US Department of Defense Security Policy: „trusted system or component“ defined as „one which can break the security policy“
- Also called “Trustworthy Computing”, “Safer Computing”... [*Trusted Computing FAQ*]



- TCPA (now TCG)
- NGSCB by Microsoft (formerly Palladium)
- Intel Trusted Execution Technology (formerly LaGrande)
- TrustZone by ARM

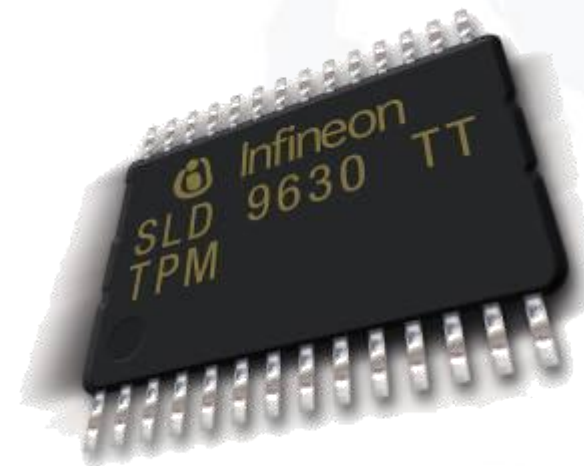
TrustZone[®]
Security Foundation by ARM[®]



- Trusted Computing Platform Alliance was founded in 1999 by Compaq, HP, IBM, Intel and Microsoft
- 2003 - Trusted Computing Group (TCG)
- Currently more than 200 members
- Changes to platform
 - Extra: Trusted Platform Module (TPM)
 - Software changes: BIOS + OS
- Main properties
 - Secure bootstrap
 - Platform attestation
 - Protected storage

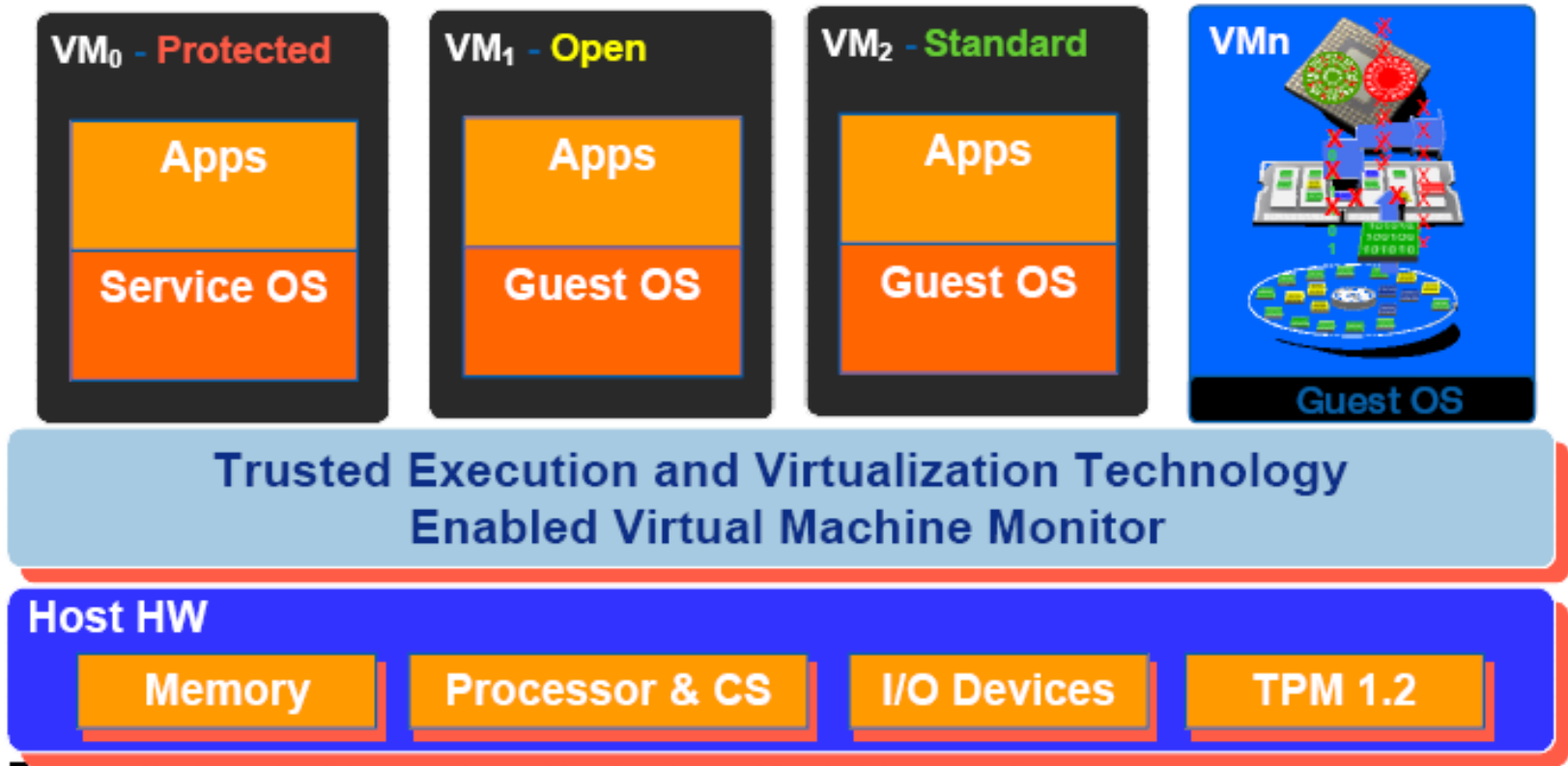
Module on the motherboard that:

- Protects secrets from attackers
- Performs cryptographic functions
 - RSA, SHA, RNG
 - Meets encryption export requirements
- Can create, store and manage cryptographic keys
- Provides a unique Endorsement Key (EK)
- Performs digital signature operations
- Holds Platform Measurements (hashes)
- Anchors chain of trust for keys, digital certificates and other credentials



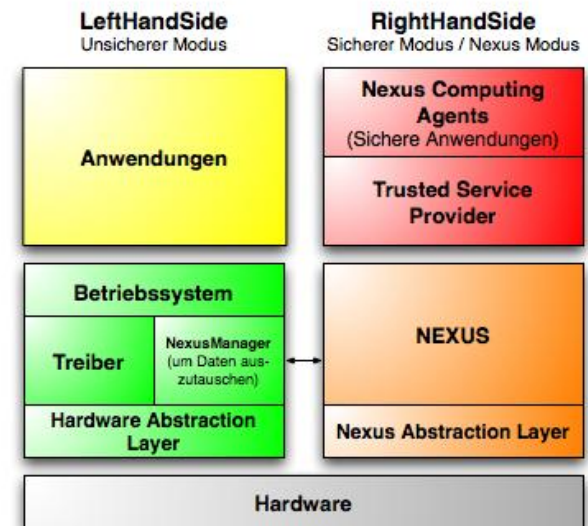
- Domain separation
- Protected execution
- Sealed storage
- Trusted channel to graphics and input devices
- Authentication of launch of a protected environment
- Attestation of platform ID

- Virtual Machine Monitor („Hypervisor“)



Next Generation Secure Computing Base

- Attestation
 - Ability to verify the operating environment
 - Remote verification
- Strong Process Isolation
 - Memory isolation (curtained memory)
- Sealed Storage
 - Data bound to operating environment
 - Application, OS, drivers, CPU, hardware, TPM
- Secure Path to I/O:
 - No keyboard sniffing
 - No frame buffer reading/writing



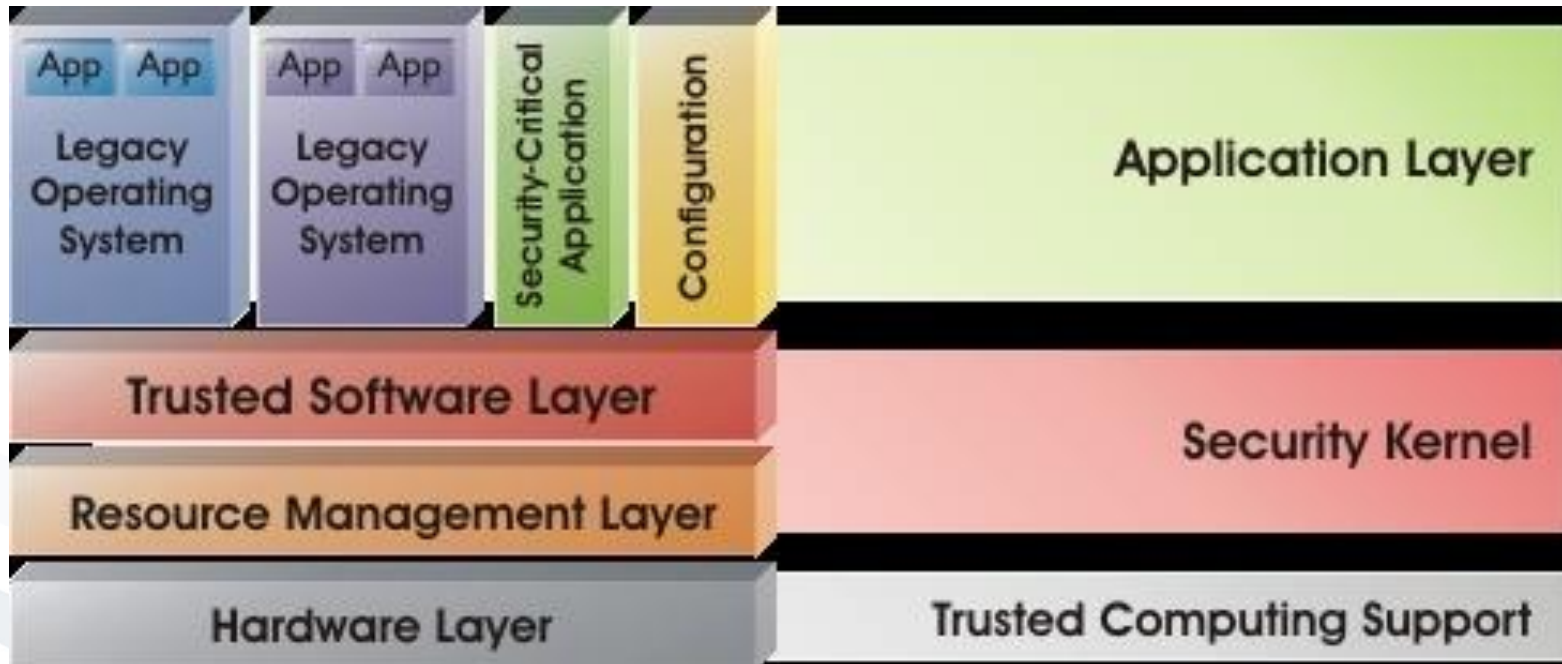
Trusted Platform Module Services

- BitLocker Drive Encryption (Secure Startup)
 - Ensures Boot Integrity
 - Resilient to Attack
 - Locks System when tampered with
 - Encrypts User Data and System Files
 - Provides Umbrella Protection for Third Party Applications
 - Simplifies the Recycling Process
 - Speeds Data Deletion

European Multilaterally Secure Computing Base (EMSCB)

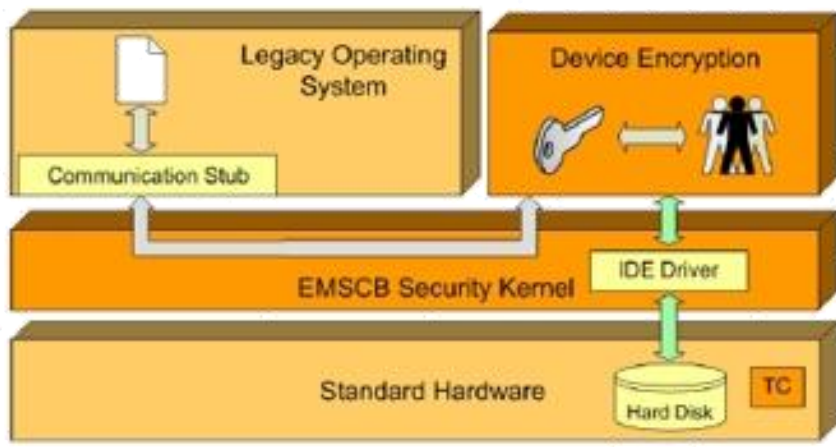
- A trustworthy computing platform
- Employing open standards
- Solving many security problems of conventional platforms.
- Hardware functionalities provided by Trusted Computing
- Security kernel based on a microkernel
- Efficient migration of existing operating systems

- Architectural Layers

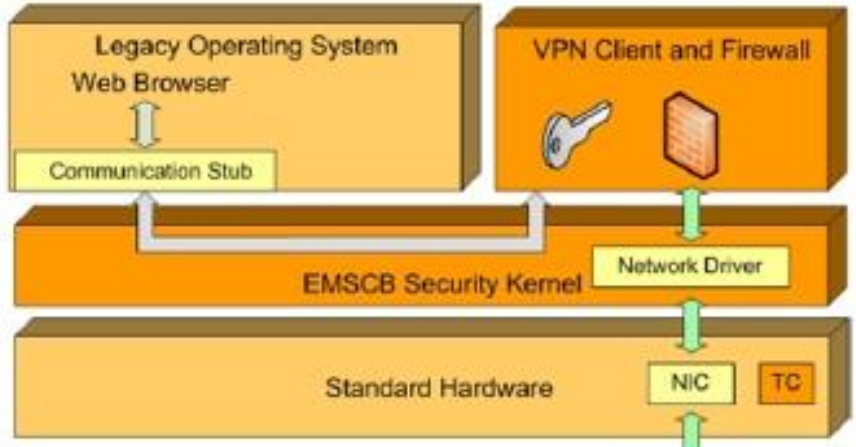


- Implementation of EMSCB architecture
- Security and access policies
- Based on PERSEUS security framework
- L4 micro kernel
- Two prototypes as proof of concept (June 2006):
 - Turaya.Crypt: Device/drive encryption, transparent for user.
 - Turaya.VPN: Secure IPSec VPN-Client compatible with conventional VPN servers.





 Unencrypted Communication
  Encrypted Communication



 Unencrypted Communication
  Encrypted Communication

- Project Trusted Mobile Platform (2003-2004)
 - developed by IBM, Intel, and NTT DoCoMo
 - a set of specifications that defines security features for mobile devices
- TCG Technical Workgroups
 - PDA Specific Implementation
 - Mobile Phone Specific Implementation
 - TCG Mobile Trusted Module Specification (2006)

[Trusted Computing Group]

- Platform Integrity
- Device Authentication
- Robust DRM Implementation
- SIMLock / Device Personalisation
- Secure Software Download
- Secure Channel between Device and UICC
- Mobile Ticketing
- Mobile Payment
- Software Use
- Proving Platform and/or Application Integrity to End User
- User Data Protection and Privacy

- **J.P. Anderson:** “*Computer Security Technology Planning Study*,” ESD-TR-73-51, Vols I and II, NTIS AD758206, Hanscom Field, Bedford, MA (October 1972).
<http://csrc.nist.gov/publications/history/ande72.pdf>
- **Matt Bishop:** *Introduction to Computer Security*. Boston: Addison Wesley, 2005. pp. 363-386.
- **Claudia Eckert:** *IT-Sicherheit*. München, Wien: Oldenbourg, 2006. pp. 37-45
- **Dieter Gollmann:** *Computer Security*. Chichester, New York, Weinheim, Brisbane, Singapore, Toronto: John Wiley & Sons, 1999. pp. 30-54
- **EMSCB -Turaya**
www.emscb.de/content/pages/About-Turaya-de.htm
- **National Information Systems Security (INFOSEC) Glossary**, September 2000
<http://handle.dtic.mil/100.2/ADA433929>
- **Intel Trusted Execution Technology**
www.intel.com/technology/security
- **ISO/IEC 2382-8, Information Technology - Vocabulary - Part 8: Security, 1998**
- **LAFKON: A movie about trusted computing; 2005; www.lafkon.net/tc**
- **LinuxDevices.com**
www.linuxfordevices.com
- **Microsoft's Next-Generation Secure Computing Base**
www.microsoft.com/resources/ngscb/default.mspx
- **Evgenia Pisko, Kai Rannenberg, Heiko Rosnagel:** Trusted Computing in Mobile Platforms - Players, Usage Scenarios, and Interests; *Datenschutz und Datensicherheit (DuD)*; 29. Jg.; H. 9, September 2005; S. 526-530
www.m-chair.net/wps/wse/dl/det/rannenberg/5674/
- **Ravi Sandhu; Xinwen Zhang.** Peer-to-Peer Access Control Architecture Using Trusted Computing Technology, SACMAT05 , Stockholm, Sweden, June 1-3, 2005
<http://portal.acm.org/citation.cfm?id=1064005>
- **Schreckxikon - Das A bis Z der Computer- und Datensicherheit**
www.sophos.de/sophos/docs/deu/papers/sophos-a-to-z-computer-and-data-security-threats.pdf
- **Trusted Computing Group**
www.trustedcomputinggroup.org/
- **Trusted Mobile Platform**
www.trustedcomputinggroup.org/developers/mobile
- **Trusted Computing FAQ**
www.cl.cam.ac.uk/~rja14/tcpa-faq.html