



BITS Pilani Presentation

BITS Pilani
Pilani Campus

Jagdish Prasad
WILP



SSZG681: Cyber Security

Lecture No: 02

Email Attacks & Operating System Security

Agenda



- The attack in web: Email attacks
 - Fake email spam
 - Phishing
 - Protecting against email attacks
- Operating system security
 - Operating system overview
 - Architecture & functions
 - Memory management functions
 - Design goals & principles
 - Layered & Kernelized design
 - Correctness and completeness
 - Secure design principles
 - Trusted systems

Email attacks



- Substantial number of emails sent everyday day are fake and can be malicious.
- Typical Email attacks are:
 - Identity theft
 - Phishing:
 - Vishing: phishing using voice communication technology
 - Smishing: phishing using text messaging on mobile platforms
 - Whaling: phishing targeting high profile persons
 - Spear phishing: phishing impersonating a trusted person
 - Pharming: impersonation of authorized website
 - Virus:
 - Spyware: collects information about user's computer activities – keyloggers, activity trackers, data capture etc
 - Scareware: persuades user to take specific action based on fear
 - Adware: Pop-up advertising message spam

Email attacks – Spear phishing

innovate

achieve

lead

TARGET

Directed toward a specific person or organization.

INTENT

Email has some form of intent; they want the target to do something.

IMPERSONATION

Trying to impersonate someone or some entity that the target trusts.



PAYLOAD

Email contains some form of payload to get the target to take the desired action.

Fake emails



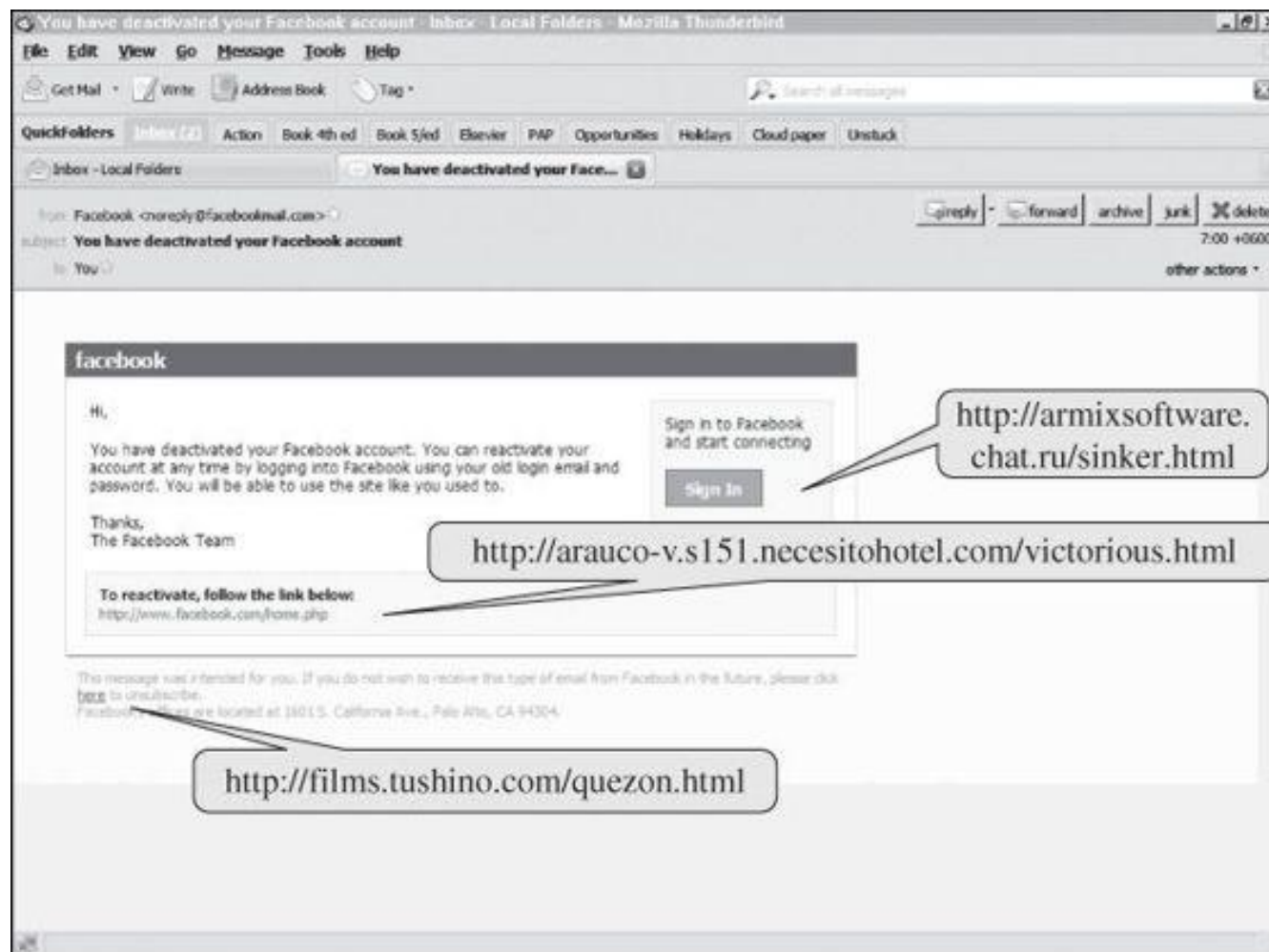
- Illegitimate emails are generated and sent by fraudsters for monetary or informational purpose
- These are done with varying degree of sophistication – some may be very poor (bad spellings, wrong English etc) but some can be very polished
- Examples:
 - Your bank account is de-activated
 - Your Facebook account is de-activated
- These ask to click on a button to activate or perform the required action. The click performs an action as desired by the fraudster
- **Motivation:** Very inexpensive and easy to send. Even if 0.1% receivers fall prey to a mail sent to 100,000, a fraudster will get 100 victims.

Fake emails : example

innovate

achieve

lead



Fake email messages as spam



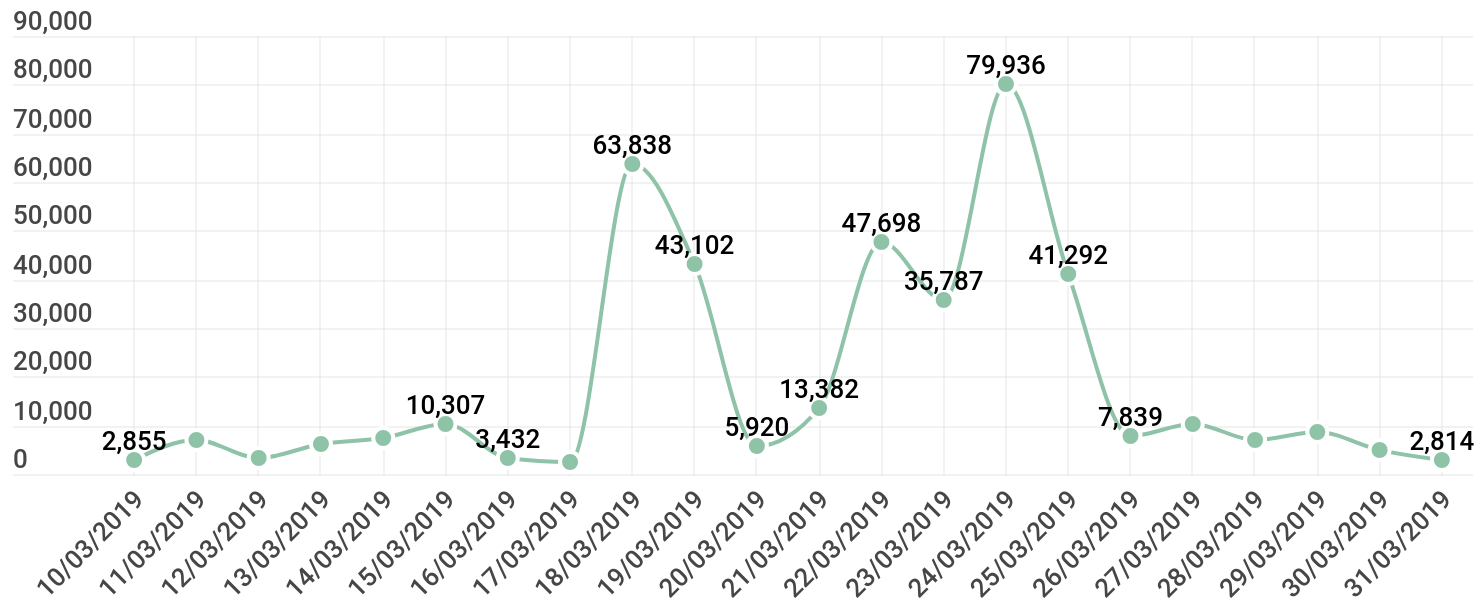
- Spam in the form of fictitious, misleading, offer to buy designer goods at throw away price, get rich schemes etc is old now
- Fake message have started using more realistic sounding subjects:
 - Fake 'non-delivery' messages ("Your message X could not be delivered")
 - False social networking messages especially attempt to obtain login details
 - Current event messages ("Want more details on event X")
 - Shipping messages ("X was unable to deliver a package to your address as shown in this link")
- Originally, emails only had static content and they would persuade a user to go to a website for action however now the action links are embedded in the mail itself – button or links

Spam volume



- M86 Security Lab estimates 86% mails are spam
- Google estimates 50-75 spam messages per user per day
- Top countries originating spam are China (22.93%), USA (19.05%), and South Korea (12.80%)
- As per Symantec's analysis of spam mails 69.7% sexual/ dating content, 17.7% pharmaceuticals, 6.2% jobs

Spam volume – Fake Apple id spams



10-Mar-2019 to 31-Mar-2019

KASPERSKY lab

Why spam



- Spammers make enough money to make the work worthwhile
- Advertising
- Pump and dump
- Publicity
- Malicious payload
- Link to malicious web sites
- Price is right – rent target addresses, pay to compose, and send message

How to stop spam

- Create Inbox filters – identify and block
- Report spam mails – mark these as fraudulent
- Don't reply or use unsubscribe option
- Unsubscribe to promotional mails (only when you are sure)
- Create a disposable email id for promotional and similar requirements
- Never reveal your email address on social media or your own website - use a 'contact me' form
- Write the address as 'a at b dot com' rather than a@b.com would evade crawlers working for spammers.
- Tag unreliable IP and email addresses
- Limit sender's email volume for a certain time period
- Take legal action against persistent criminals – Microsoft did it for Waledac
- Postage – small fee for each mail from sender (international community to agree) - [aspirational](#)

Don't invite spam

innovate

achieve

lead

Don't Invite Spam

SPAMMER TACTIC

Using spiders to crawl web pages to compile email addresses.

INBOX DEFENSES

- **Don't display your email address** on your website or social media profiles.
- **Don't open spam emails, click on the links** within the email, or **respond** to spam emails.



Fake email header data

- Original email protocol like SMTP etc are defined assuming trustworthy participants so no authentication added
- Headers in email are easy to fake like 'From:' making an email coming from a known safe source
- Email header form is standardized but not the content
- Headers like 'From:' are upto sender to define
- Check header information if in doubt. Use the feature provided by email provider like in outlook 'view source' for an email.

Email header data

innovate

achieve

lead

Message source

Received: from DM3NAM03HT098.eop-NAM03.prod.protection.outlook.com
(2603:1096:4:90::22) by SG2PR02MB3115.apcprd02.prod.outlook.com with HTTPS
via SG2PR02CA0082.APCPRD02.PROD.OUTLOOK.COM; Thu, 13 Aug 2020 07:57:21 +0000
ARC-Seal: i=2; a=rsa-sha256; s=arcselector9901; d=microsoft.com; cv=pass;
b=T09glSgaD28/hZIAPmNK7NKpMfm6JEPVd0dUsTxu9vazZIBKApXYDB5TAUTz3y5Q00pcWTBdFaWOYjuWtx//hS
H804qq17T19uQIm1U6ill4uVeFl3pbYdkm3qSkal/cRm4fWV5ouzR+ItYp3S6xxXKLQSTSjhebuglxC3FIPTnmnm69G
O0zkpHHPe93u7/eLhqCAWTmC1TgeKvgpn/BueXLnHXb9m1L8Pw11I9I4zs+GIBZTP9XPxly2UbSAsH3pf5LI+ywwN
tNDINhzRJD/2Tt1XvUzvc7iG5PysrOwKJXuz+Kga95e6YR60rppnDWYyMXS/i9tSz+mTfNjjZsQ==
ARC-Message-Signature: i=2; a=rsa-sha256; c=relaxed/relaxed; d=microsoft.com;
s=arcselector9901;
h=From:Date:Subject:Message-ID:Content-Type:MIME-Version:X-MS-Exchange-SenderADCheck;
bh=uk3ja6DR+oPgBVvGxrpFb0yq6YrYw31IPlo95qkRWPw=;
b=f0nSt7DnHwRasF8EtODn/JhzNYSbsBrhv1WicNKNfCDnuEyPfWEFJLFOJbxuqnPZhQ/NhkIK46XxV6Cdp6Sb9VzX
6PPavdjZZV5aliwS+qLm+YFEftM2o+ByiV+xM4GN68M767eNO+aQHs6MHVC4gYUS0T1NgCt8ugUVhExffFbJvH
YD30n1liVNK0sG2DHwMKRff/ZYB4bBtvsFAPQHgUNFqOX7/4Qj5mOe02dLiZEbhCln1WD1X6rSOOq4sORIf0QM
sv9oH2yi4HWhlgWwl7IXcyvKV/8sYI09++j1zouA3/KUeAZ9kDCKFVISO2Jq0GKvub32EGyNhbxTCSOGg==
ARC-Authentication-Results: i=2; mx.microsoft.com 1; spf=pass (sender ip is
40.107.236.69) smtp.rcpttodomain=hotmail.com
smtp.mailfrom=accountprotection.microsoft.com; dmarc=pass (p=reject sp=reject
pct=100) action=none header.from=accountprotection.microsoft.com; dkim=pass
(signature was verified) header.d=accountprotection.microsoft.com; arc=pass
(0 ada=1 ltdi=1 srf=[1 1 smtp.mailfrom=accountprotection.microsoft.com])

Close

Email attack types



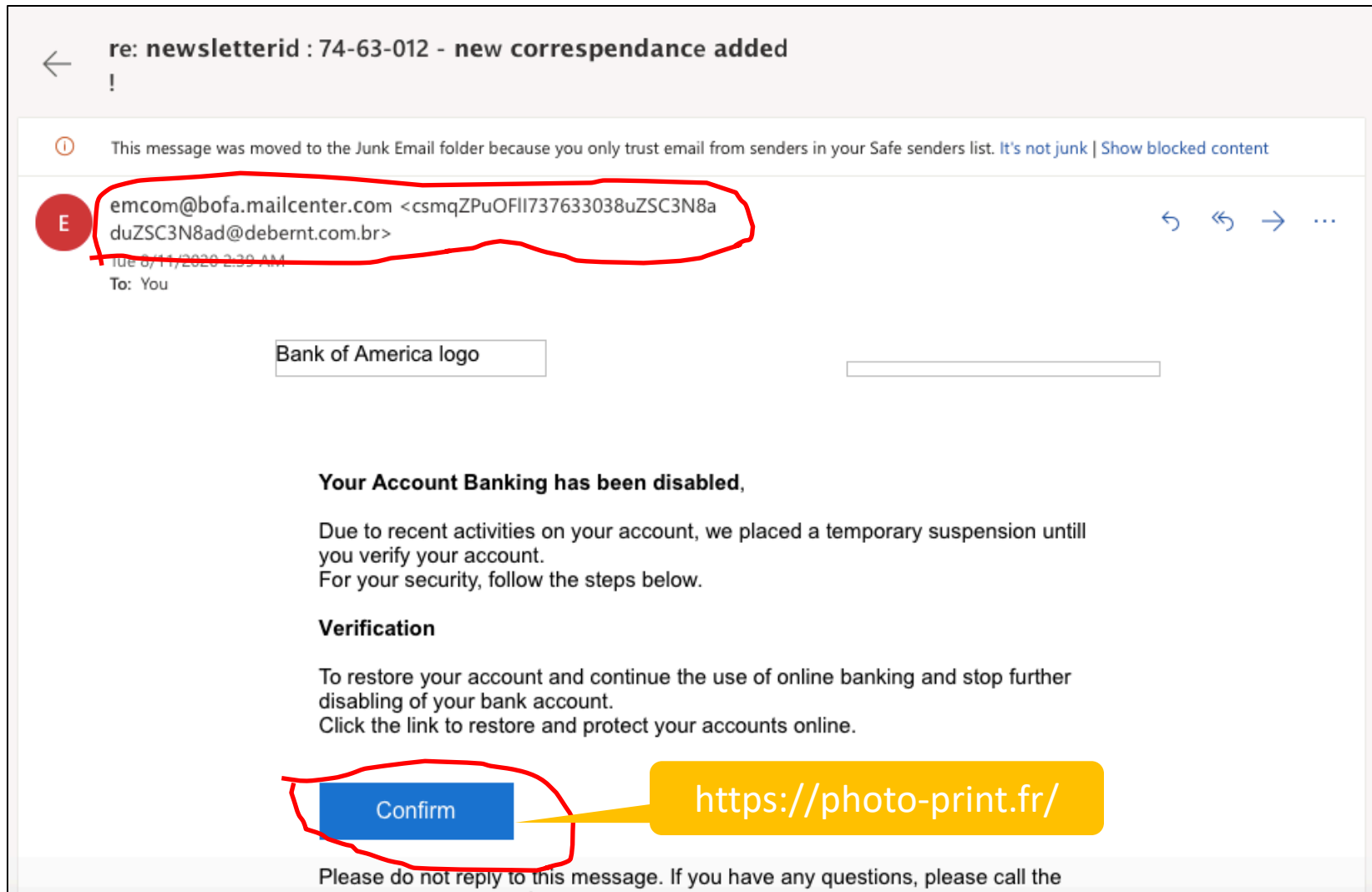
- **Phishing:** Email message tries to trick the user into disclosing private data or taking another unsafe actions. 94% receive phishing mail everyday
 - Phishing emails claim to be from trusted sources like known person, reputed companies, popular websites etc
 - Use of social engineering to personalize the message
- **Business Email Compromise (BEC):** Special form of phishing in which the attacker tricks the victim into transferring funds into the attacker's account. Often, the email will appear to be from an executive within the organization or from a legitimate vendor or business partner. A survey found that 73 percent of BEC attack victims suffered financial losses.
- **Internal Threat:** A malicious activity that spreads from one infected user to others within the organization. As per survey, 71 percent were hit by an internal threat with almost half (47 percent) hit via infected email attachments, while 40 percent said they spread through infected URLs.

Phishing techniques used by attackers



- Embedding a link in an email that redirects your employee to an unsecure website that requests sensitive information
- Installing a Trojan via a malicious email attachment or ad which will allow the intruder to exploit loopholes and obtain sensitive information
- Spoofing the sender address in an email to appear as a reputable source and request sensitive information
- Attempting to obtain company information over the phone by impersonating a known company vendor or IT department

Phishing email example



Protecting against email attacks



- Educate your employees and conduct training sessions with mock phishing scenarios. Share update on latest attack vectors/threats
- Deploy a SPAM filter that detects viruses, blank senders, etc.
- Keep all systems current with the latest security patches and updates.
- Install an antivirus solution, schedule signature updates, and monitor the antivirus status on all equipment.
- Develop a security policy that includes but isn't limited to password expiration and complexity.
- Deploy a web filter to block malicious websites.
- Encrypt all sensitive company information.
- Require encryption for employees that are telecommuting.
- Use tools or machine learning to identify and stop spoofing emails
- Analyse potential threats and isolate them

Email encryption



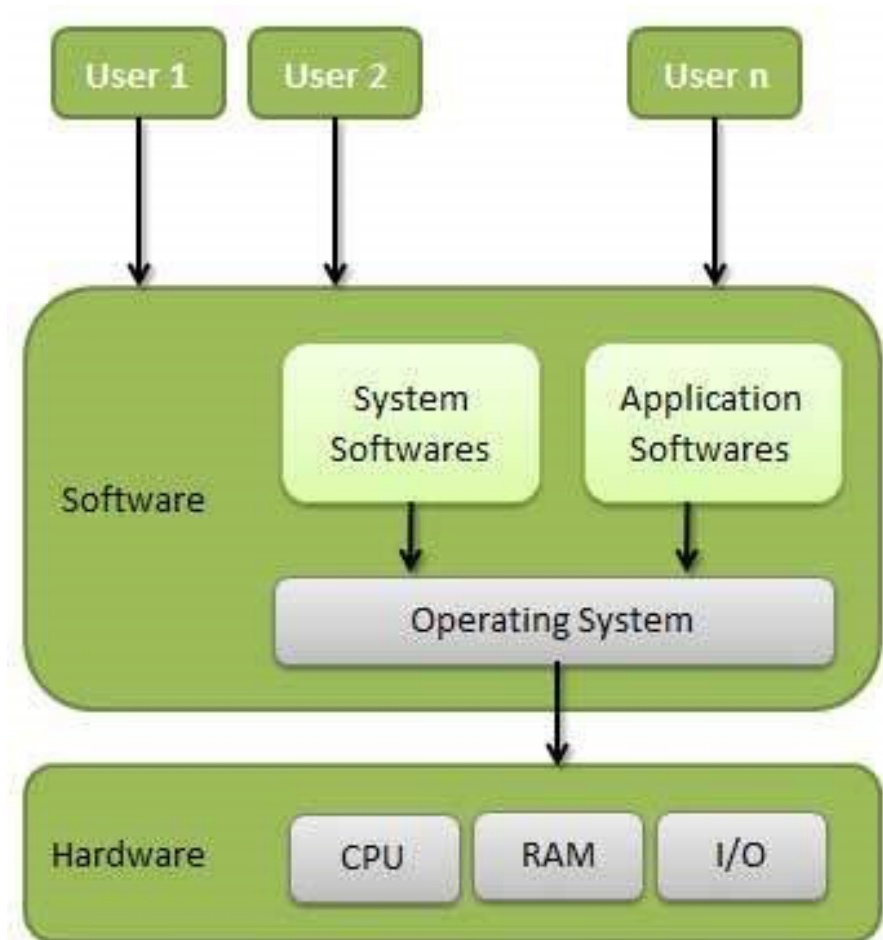
- Encryption ensures confidentiality
- PGP (Pretty Good Policy) – performs following steps
 - Create a random session key for symmetric algorithm
 - Encrypt the message using the session key (message confidentiality)
 - Encrypt the session key using recipient's public key
 - Generate a message digest or hash; sign the hash by encrypting it with sender's private key (message authenticity and integrity)
 - Attach the encrypted session key to the encrypted message and hash
 - Transmit the message to recipient
- S/MIME (Secure Multi-purpose Internet Mail Extensions)
 - Similar to PGP and is used by commercial email packages like Microsoft Outlook
 - Major difference is the method of KEY exchange
 - PGP usage Ring of Trusted while S/MIME usage hierarchy of validated certificates
 - S/MIME usage DES, AES, RC2 for encryption
 - S/MIME handles all data type like text, binary, audio, video etc in mail body as well as attachment

Operating System

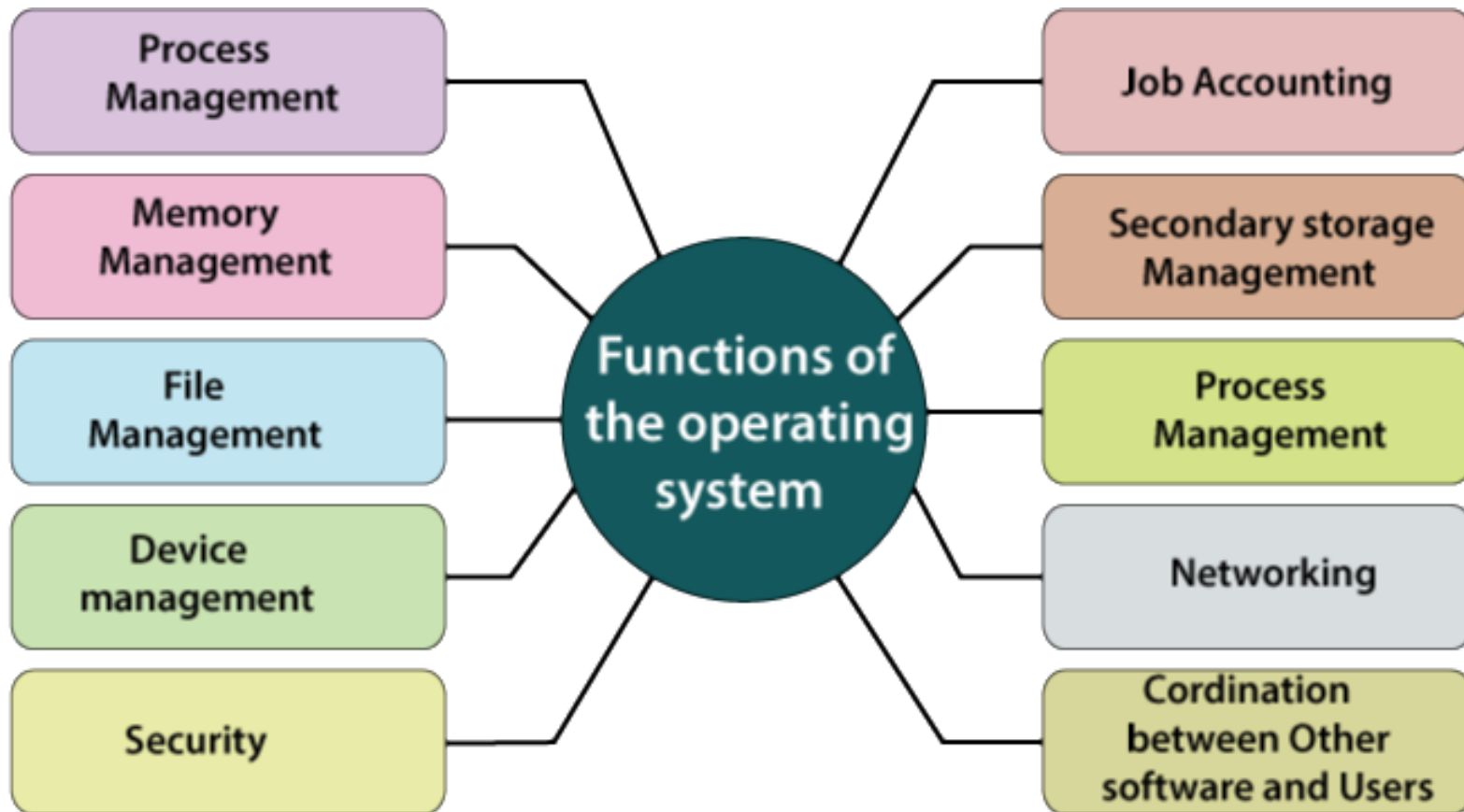
Security in operating system



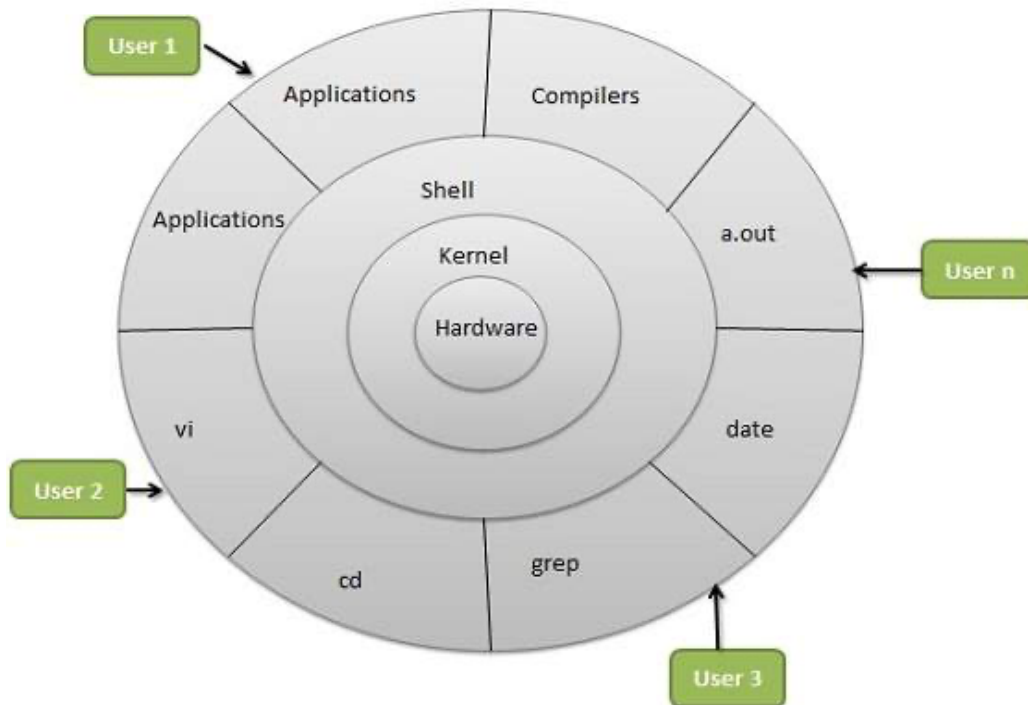
- OS is controller of all system resources which makes it primary target of attack
- OS is crucial in implementing separation and access control
- OS is initialized at boot time then initiates tasks in a sequence from disk like device drivers, process controllers, memory management etc.
- External anti-virus utilities are initiated in the last
- Any control of OS in early stage of loading will provide control on the computer



Operating system functions

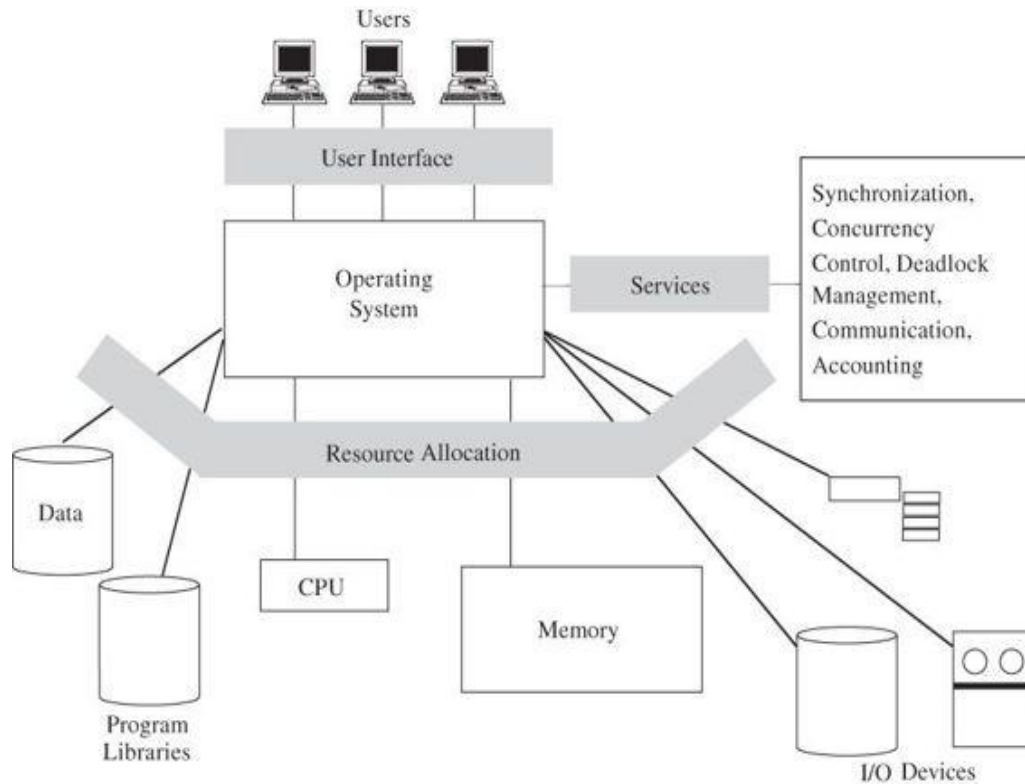


Operating system architecture



- **Hardware layer:** Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- **Kernel:** It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
- **Shell:** An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
- **Utilities:** Utility programs that provide the user most of the functionalities of an operating systems.

Operating system structure

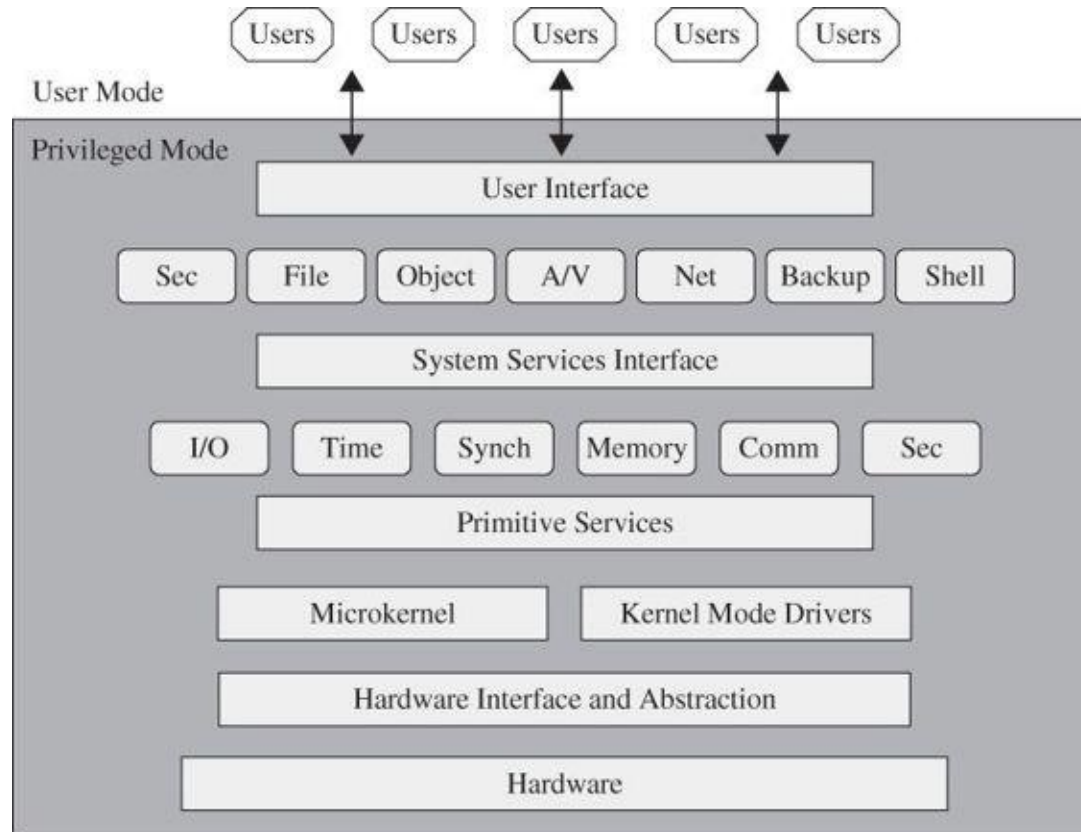


Functions involving security

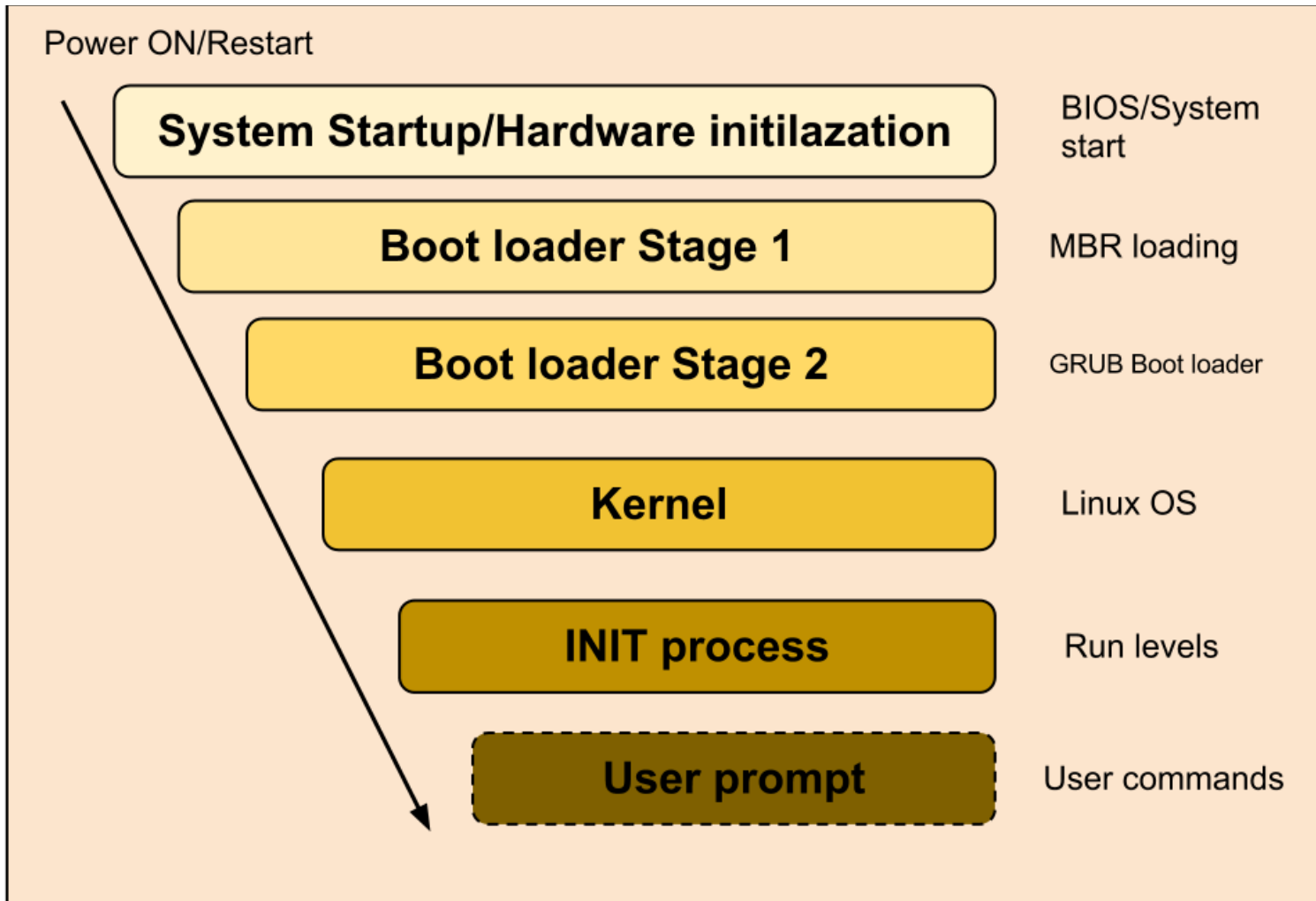
- Enforced sharing
- Inter-process communication and synchronization
- Protection of critical operating system data
- Guaranteed fair service
- Interface to hardware
- User authentication
- Memory protection
- File I/O device access control
- Allocation and access control to general objects

Fundamental functions are provided by OS Kernel

Operating system modules



Operating system loading sequence



Operating system loading sequence

BIOS	Basic input / output system executes MBR
MBR	Master Boot Record executes GRUB
GRUB	Grand Unified Bootloader executes Kernel
Kernel	Kernel executes the /sbin/init program
Init	Init executes Runlevel programs
Runlevel	Runlevel programs are executes from /etc/rc.d/rc*.d/

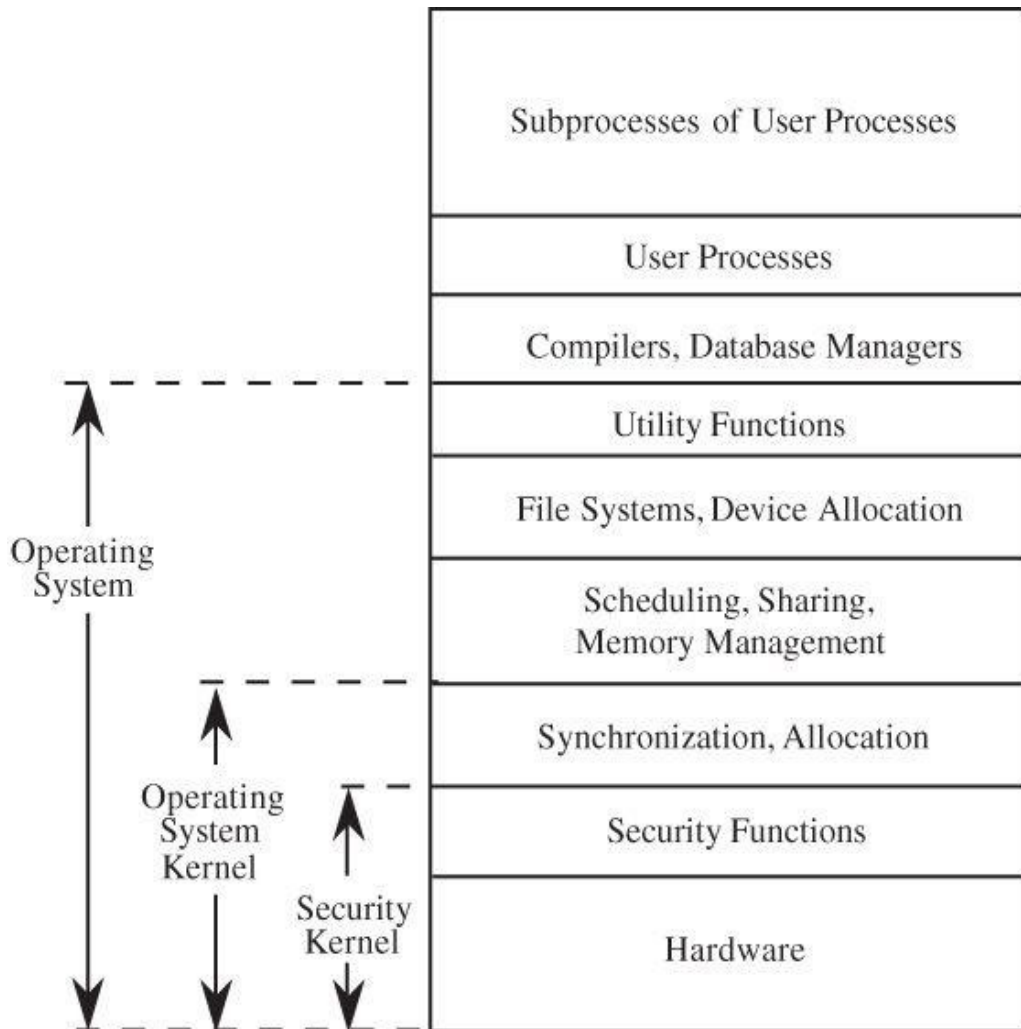
Process control block (PCB)



Process ID
State
Pointer
Priority
Program counter
CPU registers
I/O information
Accounting information
etc....

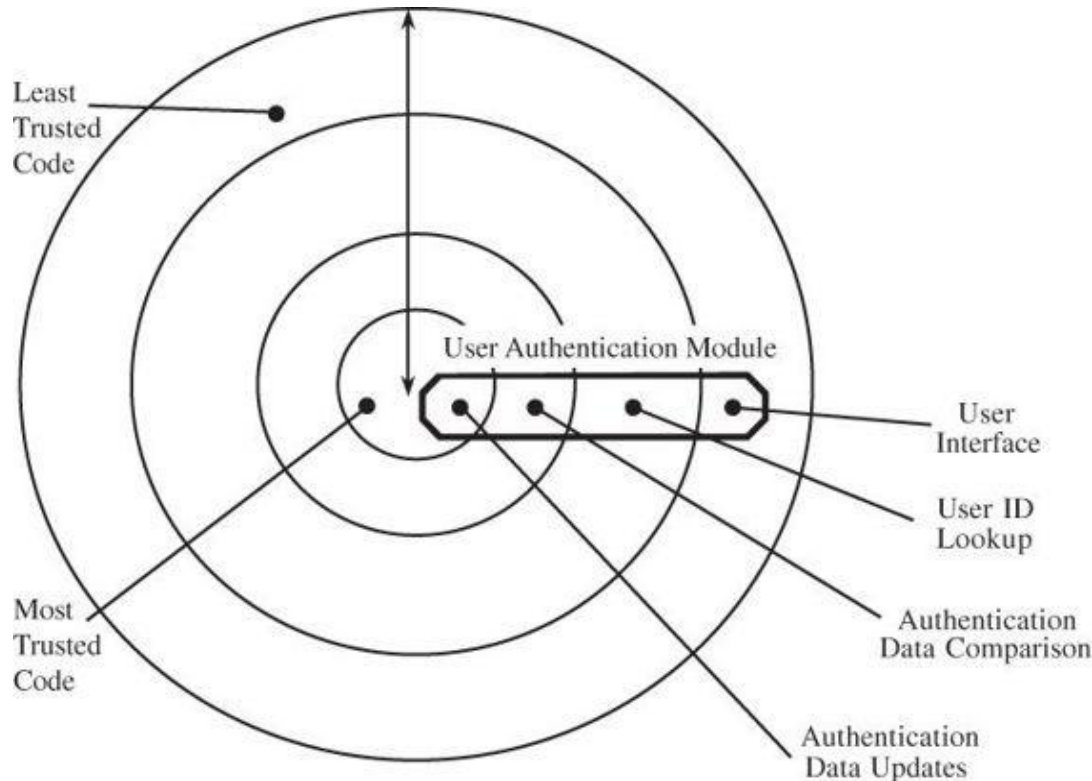
- **Process ID:** Unique identification for each of the process in the operating system.
- **Process State:** The current state of the process i.e. ready, running, waiting etc.
- **Process privileges:** Allow/disallow access to which system resources.
- **Pointer:** A pointer to parent process.
- **Program Counter:** A pointer to the address of the next instruction to be executed for this process.
- **CPU registers:** Various CPU registers where process need to be stored for execution for running state.
- **CPU Scheduling Information:** Process priority and other scheduling information required to schedule the process.
- **Memory management information:** Information of page table, memory limits, Segment table depending on memory used by the operating system.
- **Accounting information:** Amount of CPU used for process execution, time limits, execution ID etc.
- **IO status information:** I/O devices allocated to process.

Operating system layers



- Some tasks related to protection functions are performed outside the security kernel.
- Ex: User authentication may require accessing a password file, challenging the user to supply a password, verifying the correctness of the password etc.
- Disadvantage of performing all these operations inside the security kernel is that some of the operations (such as formatting the user terminal interaction and searching for the user in a table of known users) do not warrant high security.

Authentication function spanning layers in operating system



- A single logical function is implemented in several different modules
- In this design, trustworthiness and access rights are the basis of the layering.
- A single function may be performed by a set of modules operating in different layers
- The modules of each layer perform operations of a certain degree of sensitivity.

Operating system tools to implement security

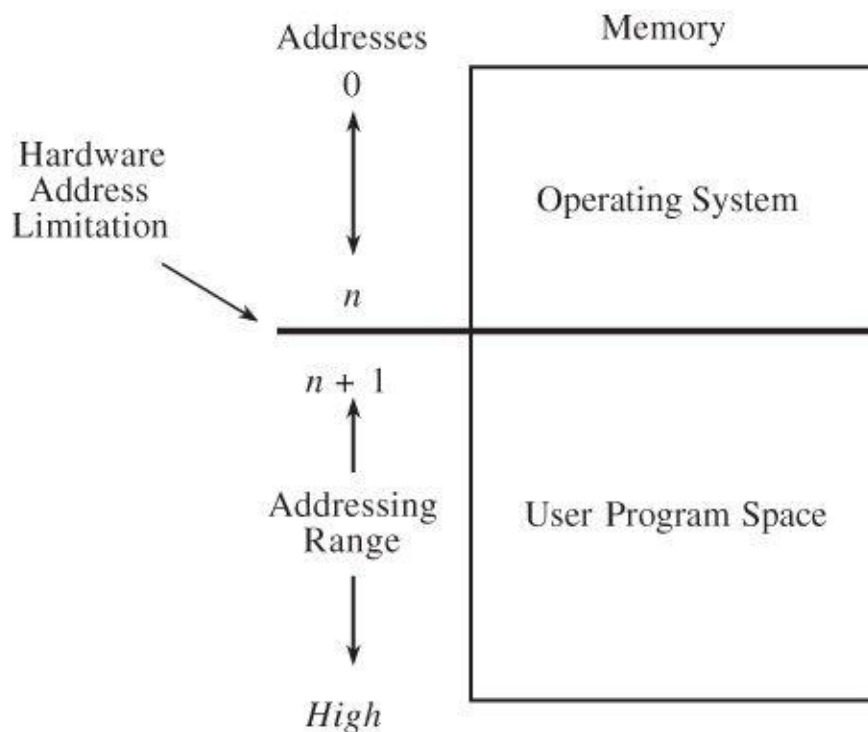


- Access control and audit log
- Virtualization: presenting a user with appearance of resource he/she is entitled to use
 - Virtual machines for each users
 - Hypervisor (Virtual machine monitor): software implementing virtual machine
 - Sandbox: an environment from which a process can have only a limited, controlled impact on outside resources
 - Honeypot: system to lure an attacker into an environment that can be both controlled and monitored
- Separation and sharing: keep one users object separate from other users objects
 - Physical separation
 - Temporal separation
 - Logical separation
 - Cryptographic separation

Hardware protection of memory



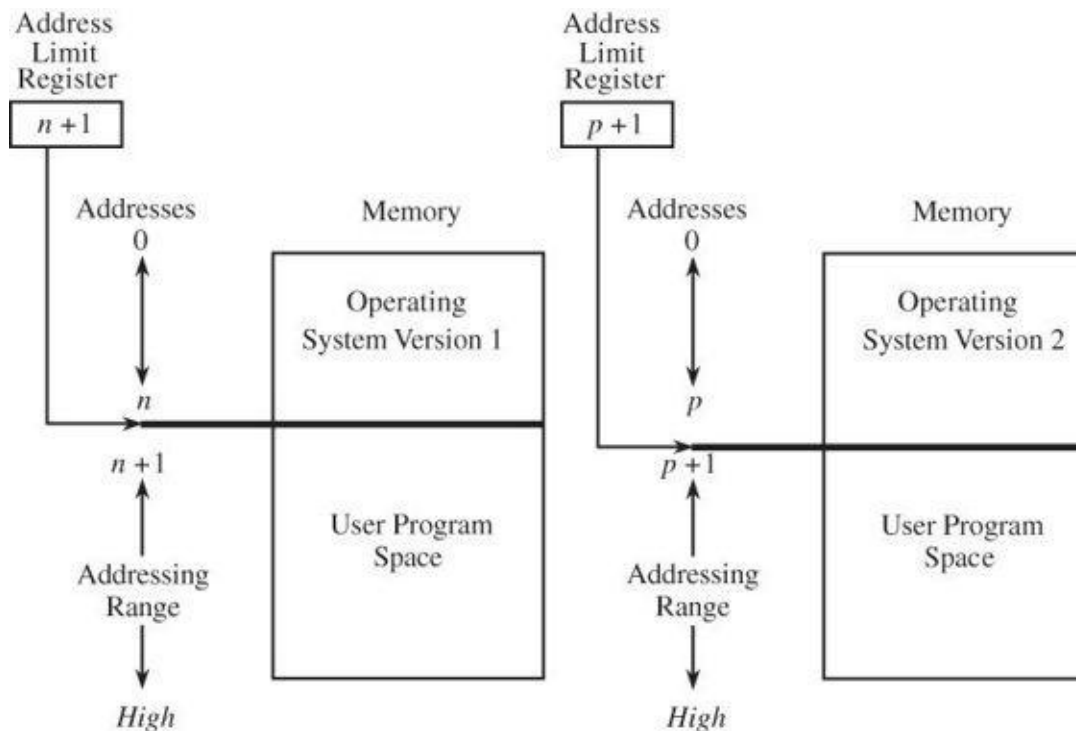
- Fence: method to confine users to one side of boundary
- Fixed fence



Fence register

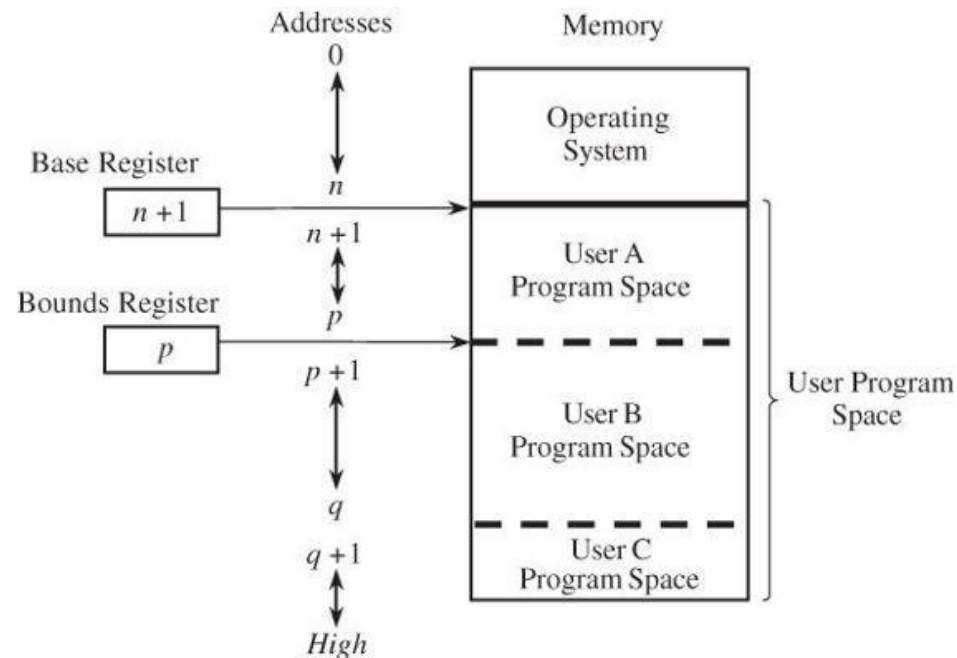


- Contains end of operating system memory boundary



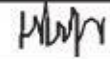

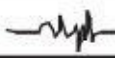



Base/Bound registers

- Base and bound registers surround a program, data area or domain
- Base register: defines the starting address for a program
- Bound register: defines the upper address limit for a program



Tagged architecture

- Base and bound registers either allow or disallow a program to make changes to an entire data block
- Tagged architecture allows each word in memory to be tagged for access rights

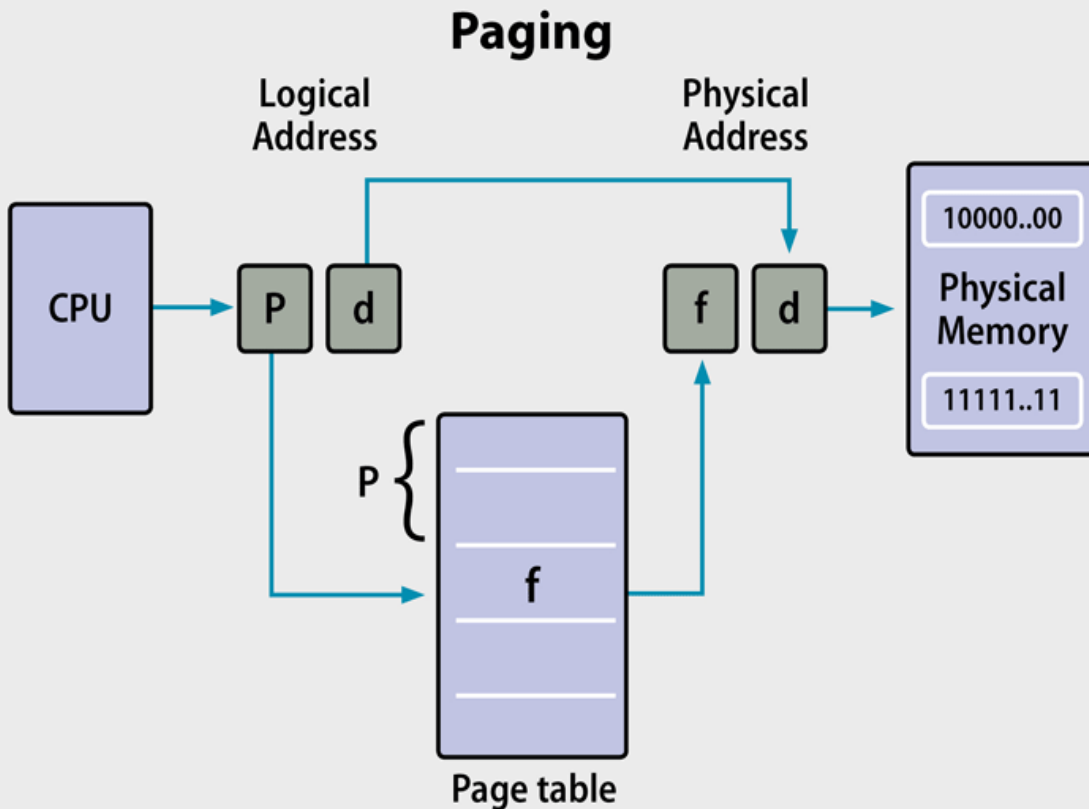
Tag	Memory Word
R	0001
RW	0137
R	0099
X	
X	
X	
X	
X	
X	
R	4091
RW	0002

Code: R = Read-only RW = Read/Write
X = Execute-only

Memory Terminology

- **Page:** A fixed-length contiguous block of virtual memory residing on disk.
- **Frame:** A fixed-length contiguous block located in RAM; whose sizing is identical to pages.
- **Physical memory:** The computer's random access memory (RAM), typically contained in DIMM cards attached to the computer's motherboard.
- **Virtual memory:** Virtual memory is a portion of an HDD or SSD that is reserved to emulate RAM. The MMU serves up virtual memory from disk to the CPU to reduce the workload on physical memory.
- **Virtual address:** The CPU generates a virtual address for each active process. The MMU maps the virtual address to a physical location in RAM and passes the address to the bus. A virtual address space is the range of virtual addresses under CPU control.
- **Physical address:** The physical address is a location in RAM. The physical address space is the set of all physical addresses corresponding to the CPU's virtual addresses. A physical address space is the range of physical addresses under MMU control.

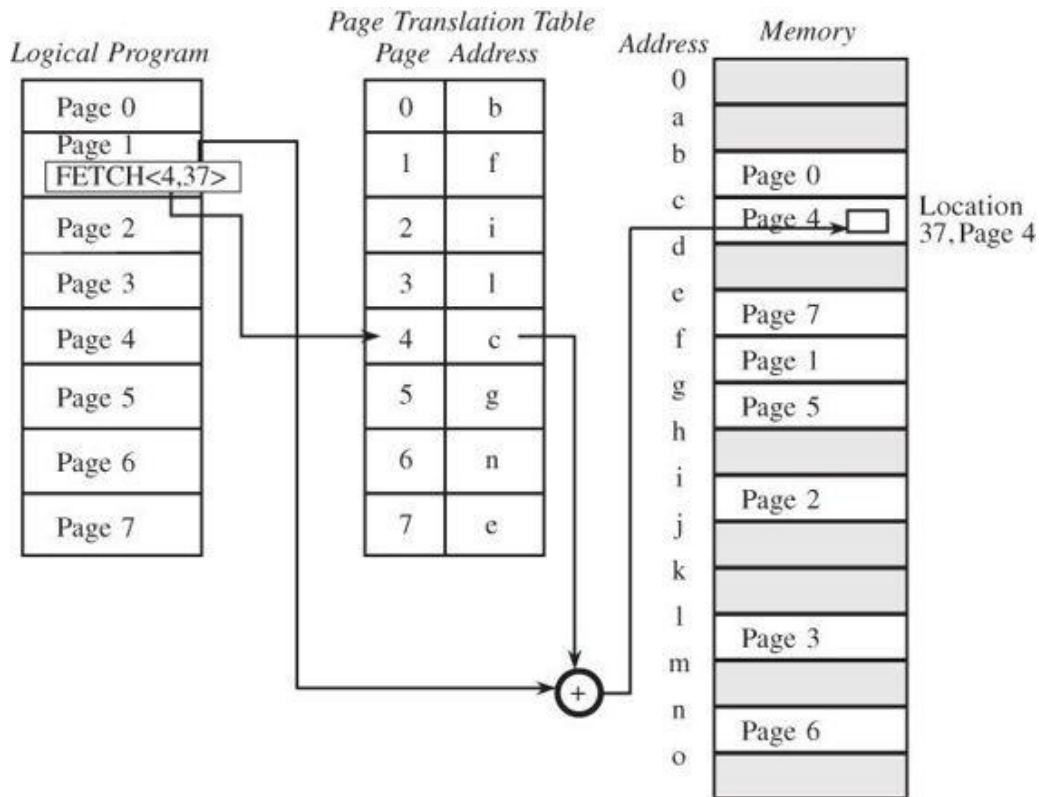
Virtual memory: Paging



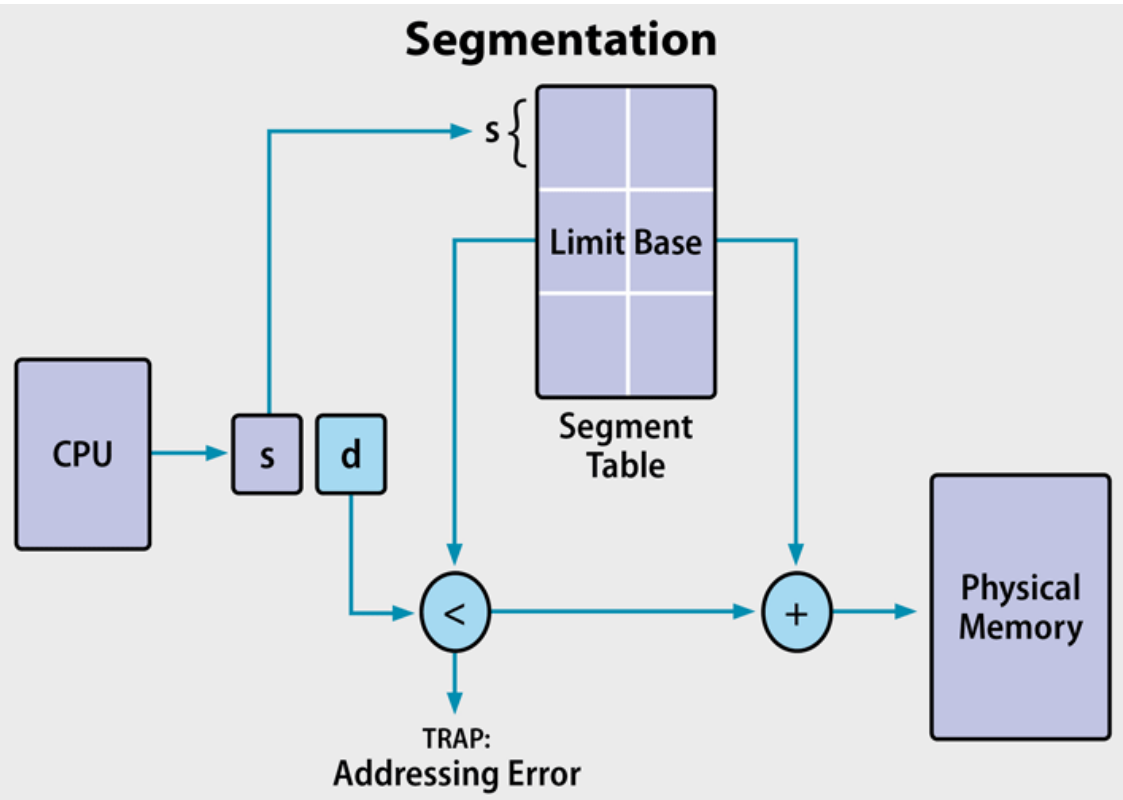
Paging specifies storage locations to the CPU as additional memory, called virtual memory. The CPU cannot directly access storage disk, so the MMU emulates memory by mapping pages to frames that are in RAM.

- A page table stores the definition of each page.
- MMU uses page tables to translate virtual addresses to physical ones.
- Each table entry indicates where a page is located: in RAM or on disk as virtual memory.
- A memory cache called the Translation Lookaside Buffer (TLB) stores recent translations of virtual to physical addresses for rapid retrieval.
- Different frame sizes are available for data sets with larger or smaller pages and matching-sized frames.

Virtual memory: Paging



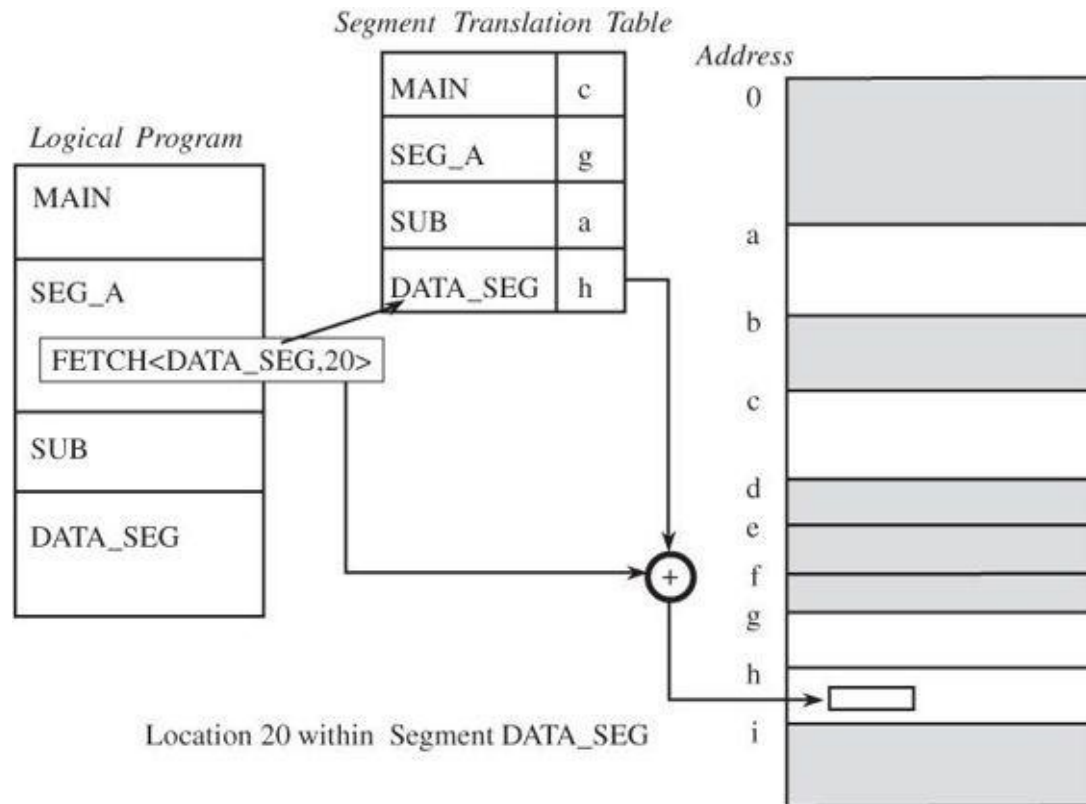
Virtual memory: Segmentation



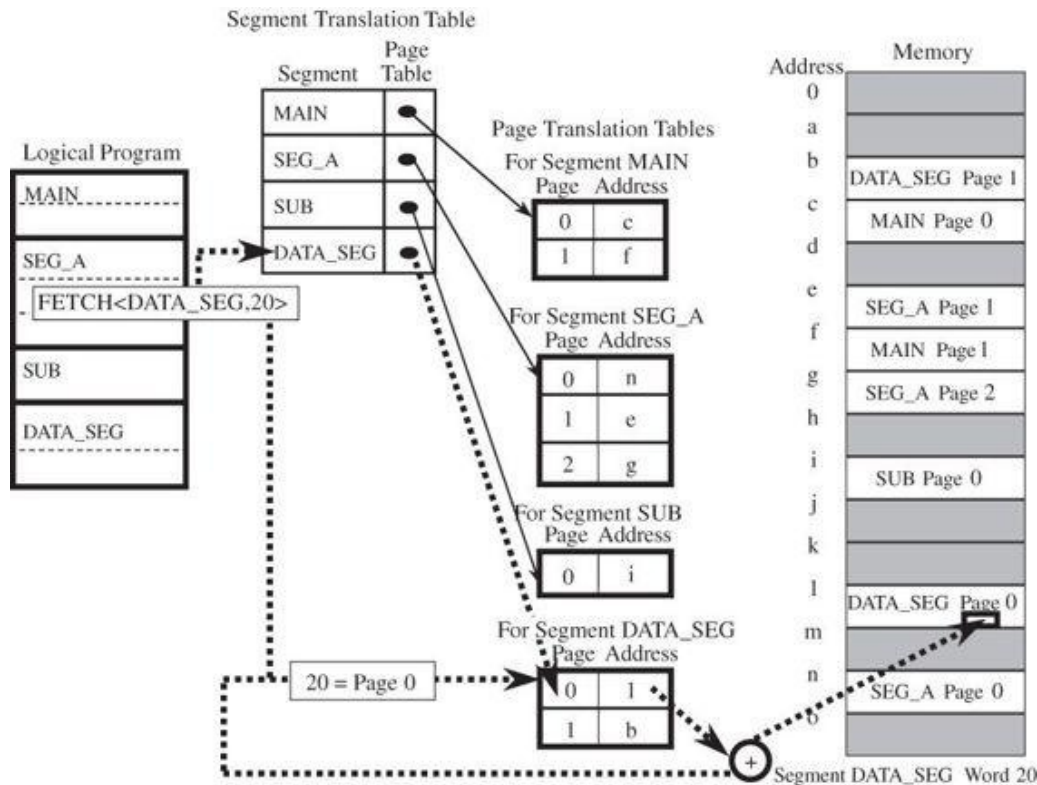
- Each segment stores the processes primary function, data structures, and utilities.
- CPU keeps a segment map table for every process and memory blocks, along with segment id and memory locations.
- CPU generates virtual addresses for running processes.
- Segmentation translates the CPU-generated virtual addresses into physical addresses that refer to a unique physical memory location.
- The translation is not strictly one-to-one: different virtual addresses can map to the same physical address.

Segmentation is a virtual process that creates address spaces of various sizes in a computer system, called segments. Each segment is a different virtual address space that directly corresponds to process objects.

Virtual memory: Segmentation



Virtual memory: Paging + Segmentation



- Modern computers use a hybrid function called segmented paging.
- Main memory is divided into variably-sized segments, which are then divided into smaller fixed-size pages on disk.
- Each segment contains a page table, and there are multiple page tables per process.
- Each of the tables contains information on every segment page, while the segment table has information about every segment.
- Segment tables are mapped to page tables, and page tables are mapped to individual pages within a segment.
- Advantages are: less memory usage, more flexibility on page sizes, simplified memory allocation, and an additional level of data access security over paging.

Operating System Security

Operating System Security



- Operating systems have old as well new piece of code
- During boot process operating system creates many open points where other pieces of functions attach during the boot process
- Exploiters find out interfaces which remain unoccupied and latch their code with that interfaces
- The more complex an operating system becomes the more chances of finding a vulnerability
- House with more windows has higher risk of being burgled than one without windows
- Simple, modular, loosely coupled design presents fewer opportunities for an attacker

Security Design Goals



- Designed for high level of protection
- Modular structure for easier control and support
- Kernel level implementation for maximum effectiveness
- Information abstraction from users
- Consistent security policy with appropriate protection
- Easy to understand, build, test and execute

Layered Design



There 4 layers of a computer system

- Hardware
- Kernel: Monolithic and Micro Kernel
- Operating system
- User
 - quasi-system programs like database managers and UI interfaces – these require separate security consideration
- Each layer has sub-layers

Layered Trust



- A secure operating system consists of series of concentric circles with most important functions at the innermost circle
- Trustworthiness and access rights of a process depend on its proximity to the centre
- Each layer properly encapsulates the functionality of layers below it
- This hierarchical structure identifies most critical parts which can be analysed intensely for correctness and security – so the number of problem areas becomes small
- Isolation limits the impact of problems to hierarchical level at or above the level of problem so harmful effects are contained

Operating System Kernel



- Part of an operating system which performs at the lowest level (nucleus or core) functions
- Implements operations such as inter process communication, synchronization, message passing and interrupt handling
- Security functions are part of security kernel (part of overall kernel)
 - Provides security interface between hardware, operating system and other parts of computer
 - Focus of all security enforcement



Security Kernel Design Consideration

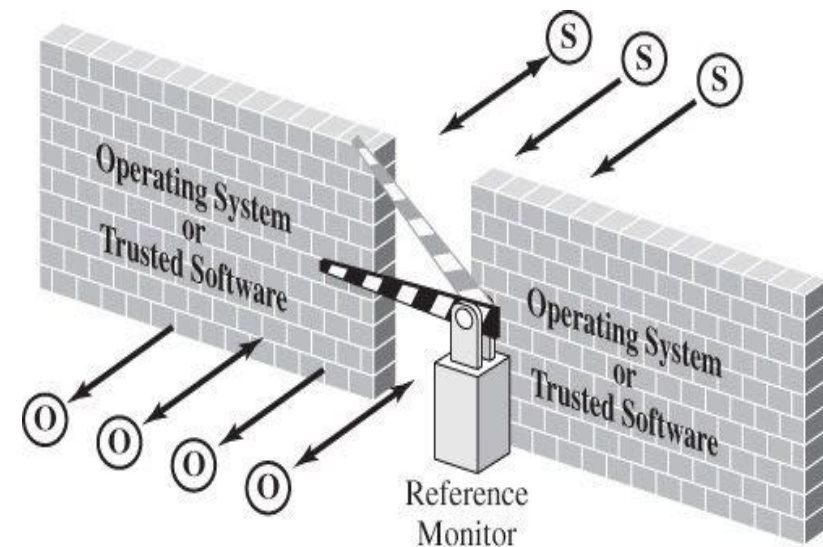
- **Coverage:** Every access to a protected object must pass through the security kernel. Security kernel can ensure that every access is checked.
- **Separation:** Isolating security mechanisms both from the rest of the operating system and from the user space makes it easier to protect those mechanisms from penetration by the operating system or the users.
- **Unity:** All security functions are performed by a single set of code, so it is easier to trace the cause of any problems that arise with these functions.
- **Modifiability:** Changes to the security mechanisms are easier to make and easier to test. And because of unity, the effects of changes are localized so interfaces are easier to understand and control.
- **Compactness:** Because it performs only security functions, the security kernel is likely to be relatively small.
- **Verifiability:** Being relatively small, the security kernel can be analyzed rigorously. Formal methods can be used to ensure that all security situations (such as states and state changes) are covered by the design.

Components of Security Module



- Reference monitor
- Authentication processing
- Identification
- Auditing
- Setting enforcement parameters

Reference Monitor



- Part of security kernel that controls access to objects
- Enforces that a subject can only access those objects which are allowed by security policy for that subject
- Controls access for devices, files, memory, inter process communication and other objects
- A brick wall around operating system or trusted software to mediate access by subjects to objects
- **Tamperproof:** impossible to weaken or disable
- **Unbypassable:** always invoked when access to any object is required
- **Analyzable:** small enough to be subjected to analysis and testing, the completeness of which can be ensured

Correctness and Completeness



- Correctness means that design clearly defines which object will be protected in what way and what subject will have access and at what level
- Completeness means that security functionality is included in all places necessary
- Security is never an add-on, it's part of initial philosophy, requirements, design, and implementation



Design Principles for Secure Systems

- **Least privilege:** Each user and each program should operate using the fewest privileges possible. This minimizes the accidental or wilful damage.
- **Economy of mechanism:** The design of the protection system should be small, simple, and straightforward. Such a protection system can be carefully analyzed, exhaustively tested, perhaps verified, and relied on.
- **Open design:** The mechanism should be public, depending on secrecy of relatively few key items, such as a password table. Public scrutiny of an open design will provide independent confirmation of the design security.
- **Complete mediation:** Every access attempt must be checked - both direct access attempts (requests) and attempts to circumvent the access.
- **Permission based:** The default condition should be denial of access.
- **Separation of privilege:** Access to objects should depend on more than one condition, such as user authentication plus a cryptographic key.
- **Least common mechanism:** Systems employing physical or logical separation reduce the risk from sharing.
- **Ease of use:** An easy to use protection mechanism is unlikely to be avoided.

Trusted Systems



- Trusted system is one which has shown a degree of trust/security that it will perform certain activities faithfully
- Features of trusted systems
 - A defined policy that details what security qualities it enforces
 - Measure and mechanism by which it enforces that security policy adequately
 - Scrutiny or evaluation to ensure that the measures and mechanisms have been selected and implemented properly
- US defense department published Trusted Computer System Evaluation Criteria (TCSEC) or orange book in late 70s. This did not reach the required acceptance level
- In 2003 Common Criteria for Information Technology Security Evaluation agreed

Trusted System Guidelines



- Orange book
 - Developed in late 1970s by US department of defense for secure computing
 - Defines a two part rating scale for trusted systems with six ratings - C1 (lowest), C2, B1, B2, B3, A1 (highest)
 - It tied features with assurance at each level
 - Strict rules limited the commercial applicability of this book
- Common criteria
 - Separation between features and assurances
 - Seven assurance levels EAL1 (lowest) thru EAL7 (highest)
 - At higher levels practices are more stringent and rigorous
 - Allowed open ended protection profiles for future products like firewalls and intrusion detection devices

Trusted Systems Characteristics



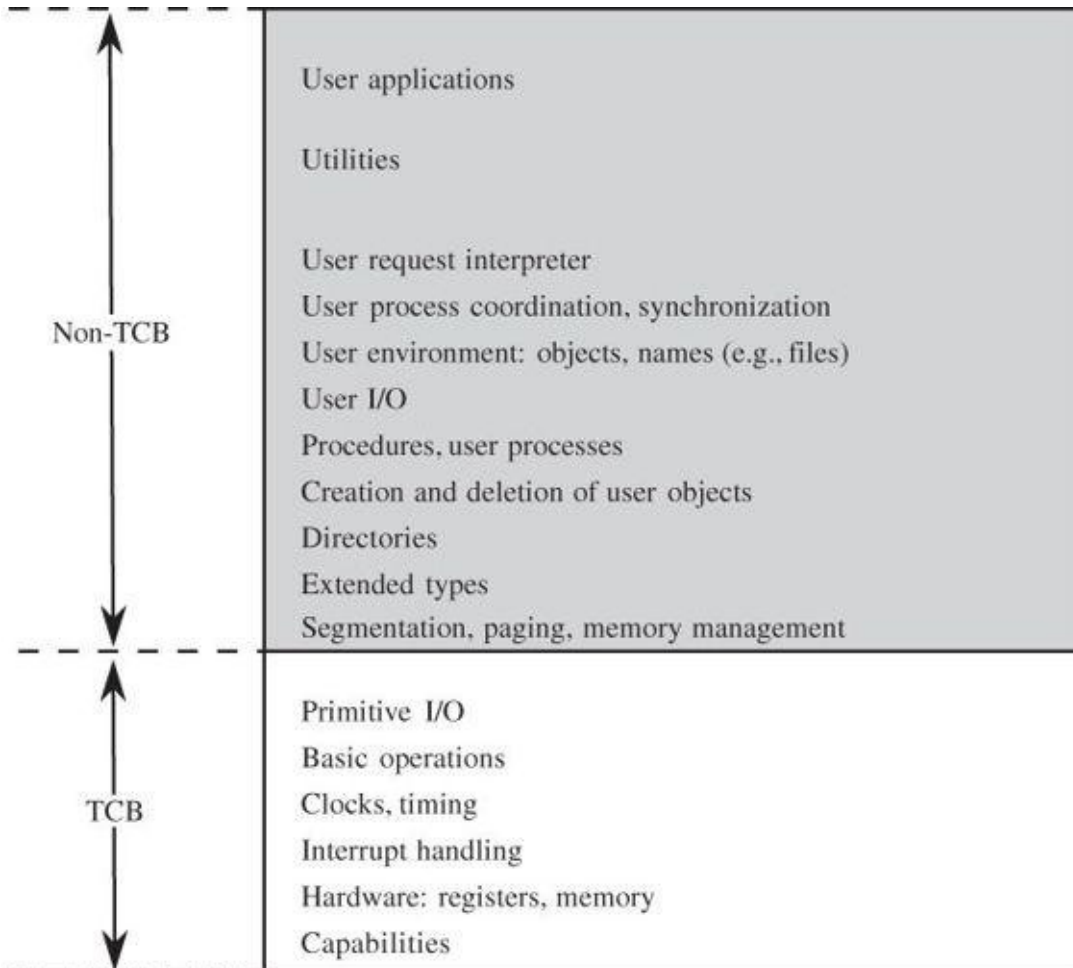
- **Functional correctness:** The program does what it is supposed to, and it works correctly.
- **Enforcement of integrity:** Even if presented erroneous commands or commands from unauthorized users, the program maintains the correctness of the data with which it has contact.
- **Limited privilege:** The program is allowed to access secure data, but the access is minimized and neither the access rights nor the data are passed along to other untrusted programs or back to an untrusted caller.
- **Appropriate confidence level:** The program has been examined and rated at a degree of trust appropriate for the kind of data and environment in which it is to be used.

Trusted System Functions



- Trusted computing base (TCB): Parts of trusted operating system responsible for correct enforcement of security policies
- TCB constituents:
 - Hardware: processors, memory, registers, clock and I/O devices
 - Security critical processes
 - Primitive files: security access control database, authentication & identification data
 - Protected memory
 - Inter-process communication

TCB & Non-TCB Sections



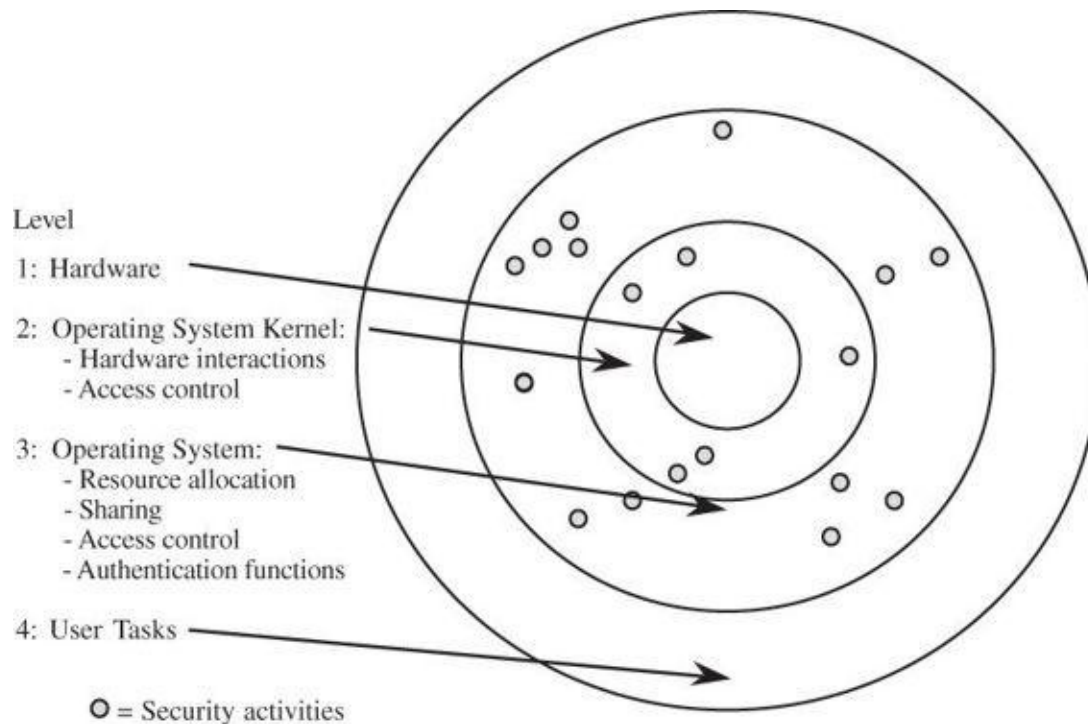
- TCB maintains secrecy and integrity of each domain
 - Process activation
 - Execution domain switching
 - Memory protection
 - I/O separation
- TCB code runs into a protected state that protects it from interference and compromise by any non-TCB code

TCB Monitored Functions



- **Process activation:** In a multiprogramming environment, activation and deactivation of processes occur frequently. Changing from one process to another requires a complete change of registers, relocation maps, file access lists, process status information, and other pointers, much of which is security- sensitive information.
- **Execution domain switching:** Processes running in one domain often invoke processes in other domains to obtain more or less sensitive data or services.
- **Memory protection:** Because each domain includes code and data stored in memory, the TCB must monitor memory references to ensure secrecy and integrity for each domain.
- **I/O operation:** In some systems, software is involved with each character transferred in an I/O operation. This software connects a user program in the outermost domain to an I/O device in the innermost (hardware) domain. Thus, I/O operations can cross all domains.

TCB Implementation



- A security kernel is built just above hardware
- Security kernel monitors all hardware access and performs all protection functions
- Secure startup is done to ensure no malicious code can block or interfere with security enforcement

Trusted Path



- A user is validated thru authentication mechanism
- A **trusted path** is an unforgeable connection by which the user can be confident of communicating directly with the operating system, not with any fraudulent intermediate application.
- A trusted path precludes interferences between a user and security enforcement mechanism of the operating system
- All security critical operations like changing a password require a trusted path between user and itself

Object Reuse



- Different user programs share computer resources
- Normally a new user writes data first and then reads
- A malicious user may claim a resource and may try to read first in which case he/she will get access to previous user data
- This attack is 'object reuse'
- Object sanitization ensures that no leakage of data happens if a subject usage an object released by another subject
- Operating systems 'clear' the object before allocating it to others

Audit



- Trusted system maintain an audit log of all security related changes like installation of new programs, or modification to operating system
- Audit log is protected against tampering, modification and deletion by unauthorized users
- Audit logging is active throughout system operations
- If audit medium fills to capacity, the system shuts down

Thank You