

Secure Operating Systems (CYB610)

Master of Science in Cybersecurity

Lecture 1 Security Threats to OS

Dr. Ibrahim Gomaa

*Faculty of Informatics
Midocean University*



Chapter

01

Topics

This chapter covers the following topics and concepts:

- **What the origins of Linux are**
- **How security works in the open-source world**
- **What distributions of Linux exist**
- **What the C-I-A triad is**
- **How Linux operates as a security device**
- **What Linux's place in the enterprise is**
- **What some examples of recent security issues are**

Chapter

01

Goals

When you complete this chapter, you will be able to:

- **Describe the basics of security in an open-source world**
- **Explain various roles of Linux systems in the IT architecture**
- **Differentiate between Linux and the operating environment that runs on top of Linux**
- **Explain threats that can target Linux**

Chapter

01

Learning Key Concepts

Key Concepts

- The origins of Linux
- Open-source software security considerations
- The confidentiality, integrity, and availability (C-I-A) triad
- Linux distributions
- Linux security and security threats

Computer Security

- **Computer security** is about provisions and policies adopted to protect **Hardware, Software, information and property** from theft, corruption, or natural disaster
 - while allowing the information and property to remain accessible and productive to its intended users.
- security of computers against intruders (e.g. Attackers) and malicious software (e.g., viruses).

Assets



Hardware:

- Computer
- Devices (disk drives, memory, printer)
- Network gear

Software:

- Operating system
- Utilities (antivirus)
- Commercial applications (word processing, photo editing)
- Individual applications

Data:

- Documents
- Photos
- Music, videos
- Email
- Class projects

Why need security at the OS level?

- No more standalone computer system environments.
- Any system can be globally accessible through a set of vast inter and intra-network connections.
- Transition motivated by the need to work remotely,
 - convenience in accessing personal records, online shopping etc.
- A single security vulnerability in the OS design and implementation known to a malicious attacker could do serious damage.

Who are the attackers?

- Vandals (Hackers, crackers) driven by intellectual challenge.
- Insiders: employees or customers seeking revenge or gain informal benefits
- Natural disasters: flooding, fire, storms, earthquake...
- Criminals seeking financial gain.
- Organized crime seeking gain or hiding criminal activities.
- Organized terrorist groups or nation states trying to influence national policy.
- Foreign agents seeking information (spying) for economic, political, or military purposes.
- Tactical countermeasures intended to disrupt military capability.
- Large organized terrorist groups

What are the vulnerabilities?

- **Physical** vulnerabilities (Eg. Computer can be stolen)
- **Natural** vulnerabilities (Eg. Earthquake)
- **Hardware and Software** vulnerabilities (Eg. Failures)
- **Media** vulnerabilities (Eg. Hard disks can be stolen)
- **Communication** vulnerabilities (Ex. Wires can be tapped)
- **Human** vulnerabilities (Eg. Insiders)
- Poorly chosen passwords
- Software bugs (non reliability of software)
 - buffer overflow attacks

What are the vulnerabilities?

- Automatically running active content: active-x, scripts, Java programs (applet)
- Open ports: telnet, HTTP
- Incorrect configuration
 - file permissions
 - administrative privileges
- Untrained users/system administrators

Security features in Operating System

- An operating system manages and controls access to hardware components
- Older operating systems focused on ensuring data confidentiality
- Modern operating systems support four basic functions
 - Positively identify a user
 - Restrict access to authorized resources
 - Record user activity
 - Ensure proper communications with other computers and devices (sending and receiving data)

Security features in ordinary OS

- Authentication of users
 - password comparison
- Protection of memory
 - user space, paging, segmentations
- File and I/O device access control
 - access control matrix
- Allocation & access control to general objects

Security features in ordinary OS

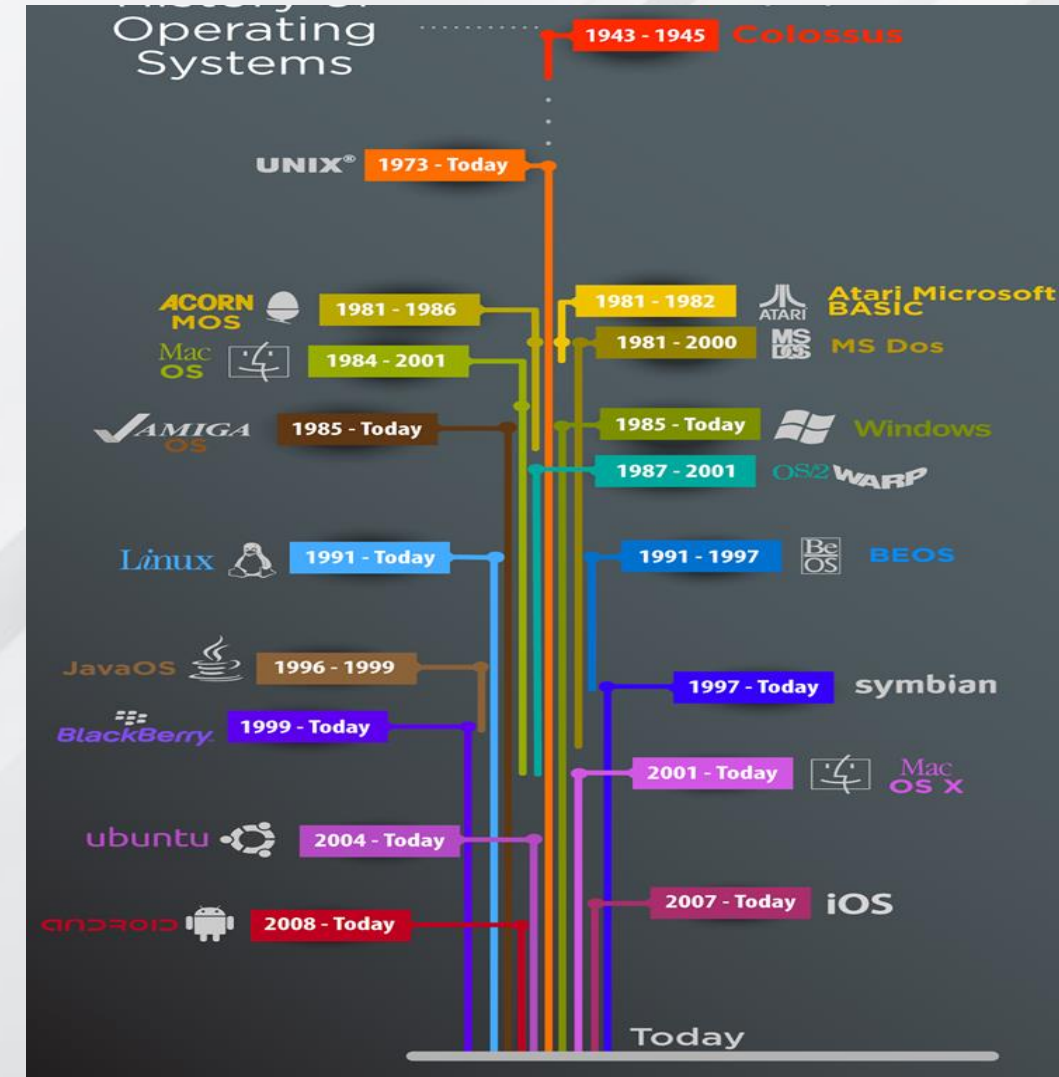
- **Enforcement of sharing resources**
 - To preserve integrity and consistency (critical section)
- **Fair service**
 - no starvation and deadlock
- **Inter-process communication & synchronization**
 - Shared variable (e.g, using semaphores)
- **Protection of data**
 - encryption, isolation

Security features in ordinary OS

- Identification and Authentication
- Mandatory Access Control (enforce multilevel security by classifying the data and users into various security classes)
- Discretionary Access Control (grant privileges to users)
- Object use and reuse protection (Subject and object)
- Anti-virus scan
- Accountability and Audit (security log)
- Firewall
- Intrusion detection (patterns of normal system usages, anomalies)

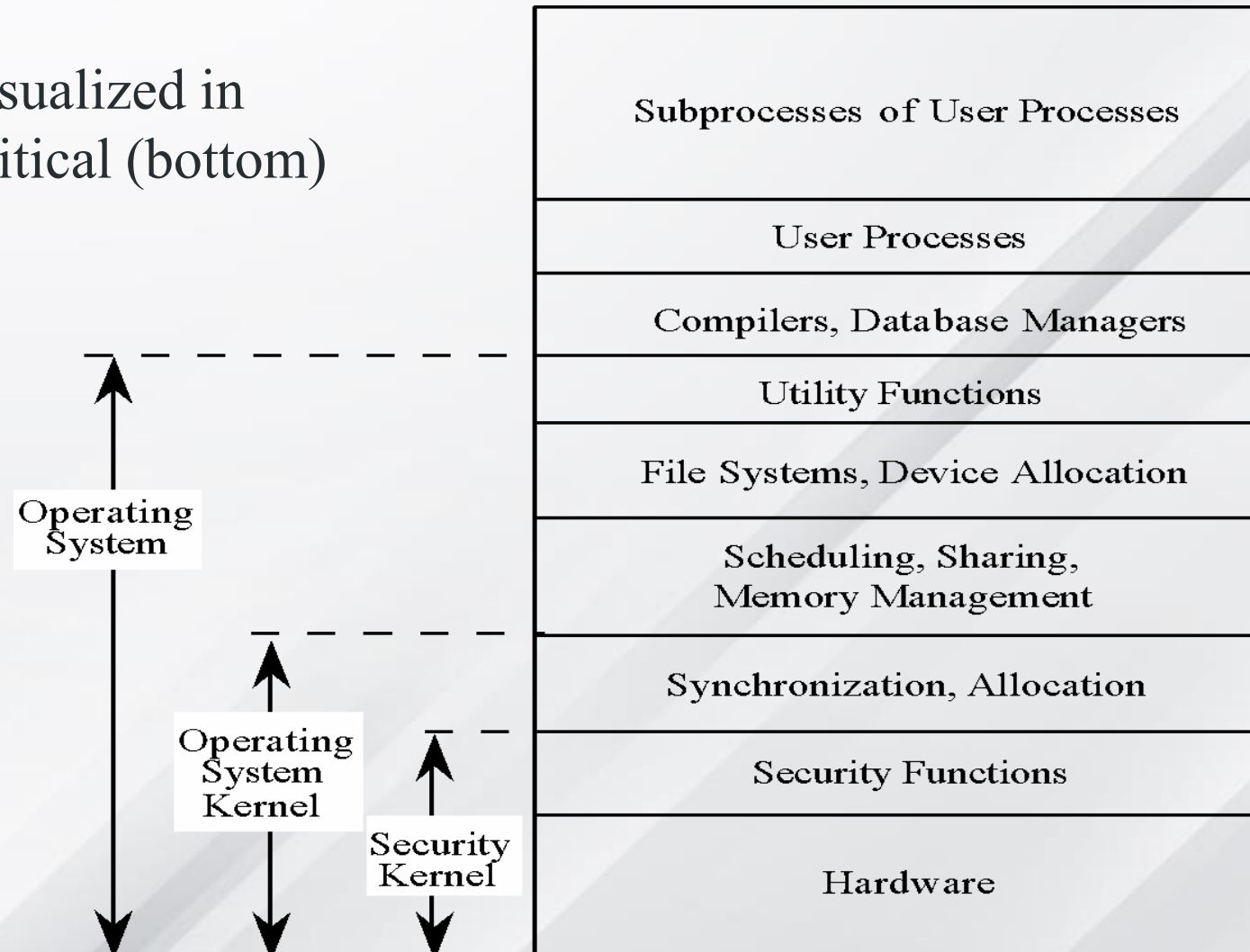
History of Operating Systems

- First, an entire computer was dedicated to one program at a time, but this approach proved wasteful
- The first operating systems saved startup, loading, and shutdown time and made much better use of limited resources
- The first personal computers took a major step back, as they were dedicated to single users and effectively one program at a time
- Multitasking returned to the mainstream in the 1990s, and with it came all the lessons of the early shared computers

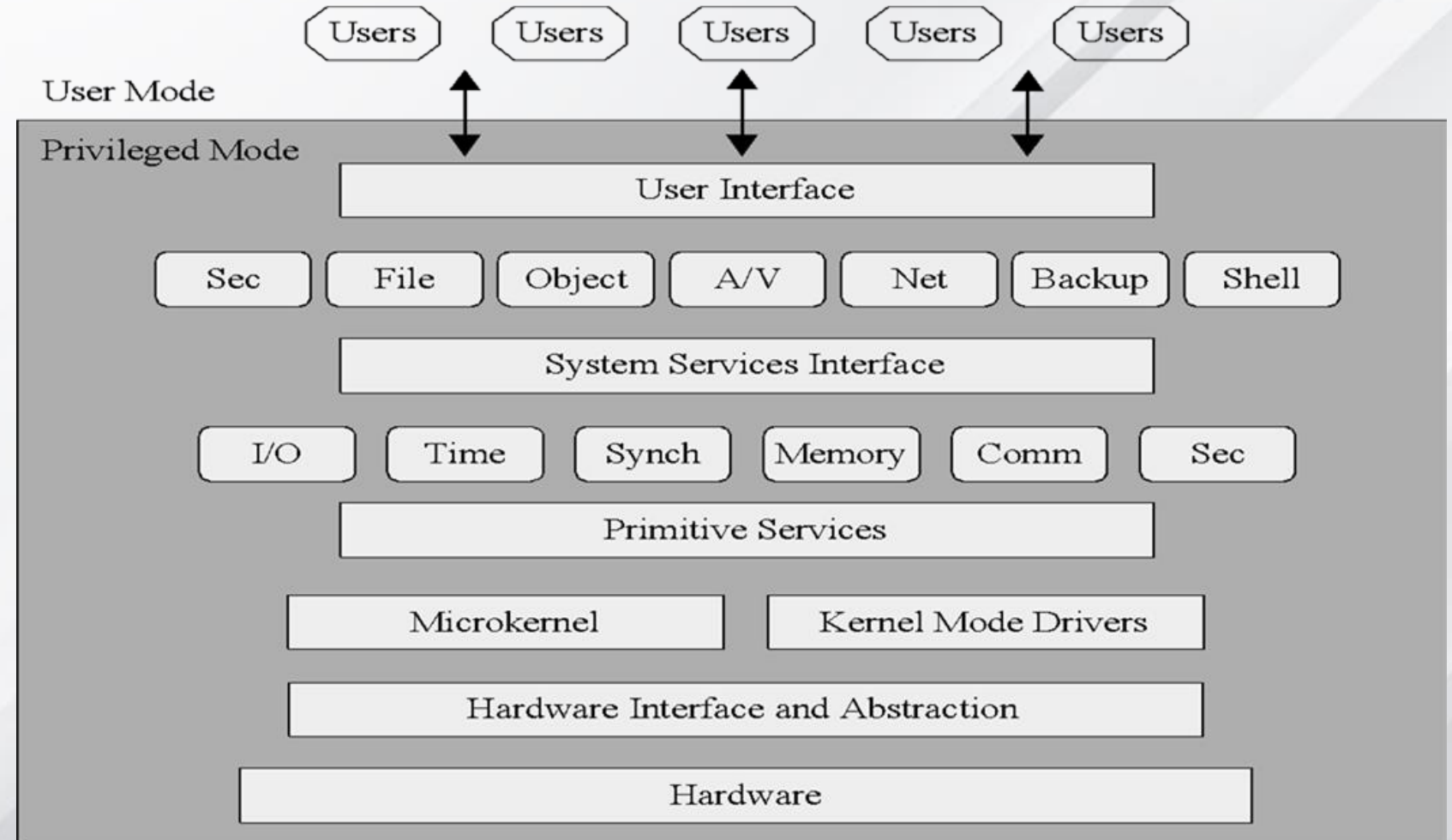


OS Layered Design

Operating system visualized in layers, from most critical (bottom) to least critical.



Modern OSs are **built from discrete modules**. These modules generally come from a variety of sources and are subject to updating/overwriting, so they cannot trust one another.



Protected Objects

These are some of the common objects that need protection by and in OSs.

- Memory
- Sharable I/O devices, such as disks
- Serially reusable I/O devices, such as printers
- Sharable programs and sub procedures
- Networks
- Sharable data

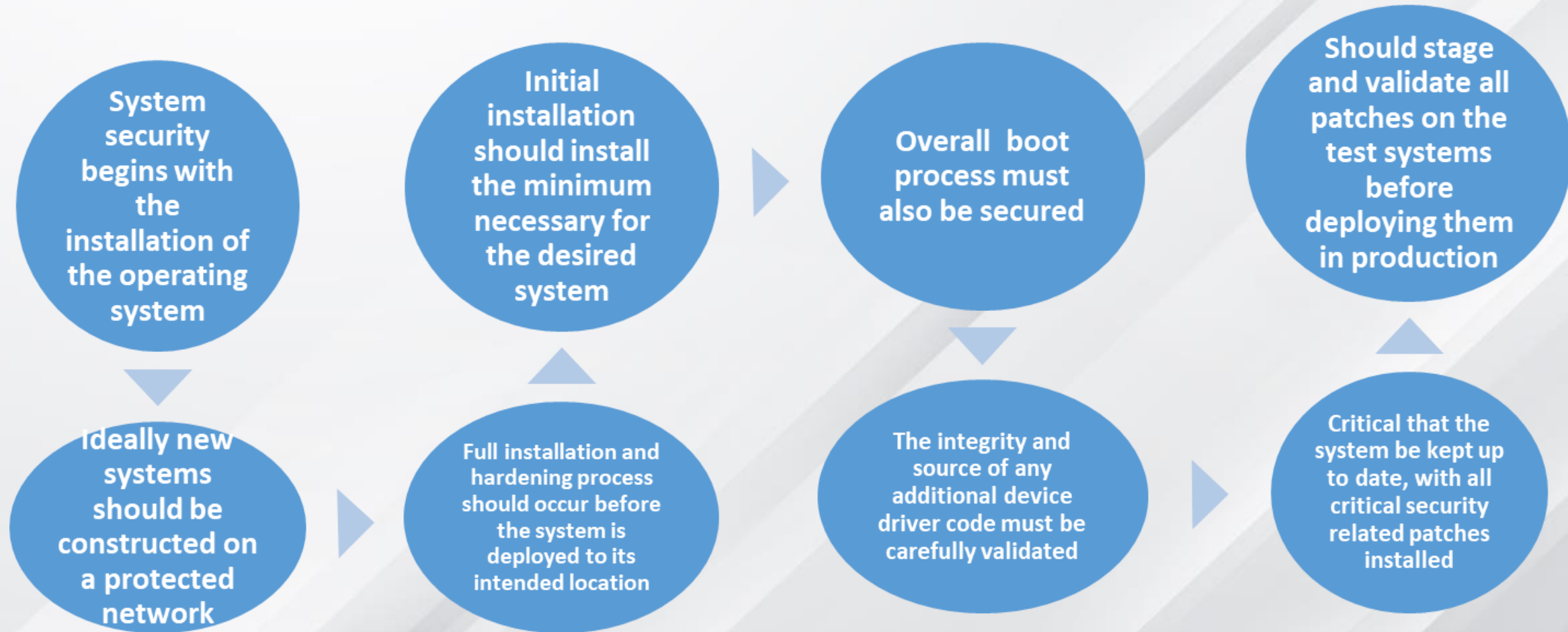
Operating Systems Hardening

➤ First critical step in securing a system is to secure the base operating system

- Basic steps

- Install and patch the operating system
- Harden and configure the operating system to adequately address the identified security needs of the system by:
 - Removing unnecessary services, applications, and protocols
 - Configuring users, groups, and permissions
 - Configuring resource controls
- Install and configure additional security controls, such as anti-virus, host-based firewalls, and intrusion detection system (IDS)
- Test the security of the basic operating system to ensure that the steps taken adequately address its security needs

Operating Systems Hardening



Operating Systems Hardening

- **System planning process should consider:**
 - Categories of users on the system
 - Privileges they have
 - Types of information they can access
 - How and where they are defined and authenticated
- **Default accounts included as part of the system installation should be secured**
 - Those that are not required should be either removed or disabled
 - Policies that apply to authentication credentials configured

Application Configuration

- **May include:**
 - Creating and specifying appropriate data storage areas for the application
 - Making appropriate changes to the application or service default configuration details
- **Some applications or services may include:**
 - Default data, Scripts and User accounts of particular concern with remotely accessed services such as Web and file transfer services
 - Risk from this form of attack is reduced by ensuring that most of the files can only be read, but not written, by the server

Security Maintenance

- Process of maintaining security is continuous
- Security maintenance includes:
 - Monitoring and analyzing logging information
 - Performing regular backups
 - Regularly testing system security
 - Using appropriate software maintenance processes to patch and update all critical software, and to monitor and revise configuration as needed

Can only inform you about bad things that have already happened

In the event of a system breach or failure, system administrators can more quickly identify what happened

Key is to ensure you capture the correct data and then appropriately monitor and analyze this data

Information can be generated by the system, network and applications

Range of data acquired should be determined during the system planning stage

Generates significant volumes of information and it is important that sufficient space is allocated for them

Automated analysis is preferred

Logging

Performing regular backups of data is a critical control that assists with maintaining the integrity of the system and user data

May be legal or operational requirements for the retention of data

Backup

The process of making copies of data at regular intervals

Archive

The process of retaining copies of data over extended periods of time in order to meet legal and operational requirements to access past data

Needs and policy relating to backup and archive should be determined during the system planning stage

Kept online or offline

Stored locally or transported to a remote site

Backup & archive

The Origins of Linux

- 1960s
 - 1964: MIT joins with GE and Bell Labs to create a multiuser, time-sharing operating system
- 1980s
 - Multiple Unix versions available from AT&T, University of California .
 - 1987: Unix-like operating system called MINIX released
- 1990s
 - 1991: Torvalds releases Linux in October
 - Numerous open-source projects contributed to Linux over past 20 years.

The Origins of Linux

- Linux is a common operating system used for Internet infrastructure.
- With Linux, security is much more than just firewalls and permissions.
- As with any operating system, a Linux implementation requires strategies to harden it against attack.

Linux Security

- The reality is that no operating system can be called secure.
- Security results from appropriate controls and processes
- Has evolved into one of the world's most popular operating systems
- Free, Open-sourced
- Available in a wide variety of distributions targeted at almost every usage scenario imaginable

Linux Security

The traditional Linux security model can be summed up quite shortly: People or processes with “root” privileges can do anything; other accounts can do much less

From the attacker’s perspective the challenge in cracking a Linux system is gaining root privileges

Once an attacker gains root privileges they can:

- Erase or edit logs & hide their processes, files, and directories
- Basically, redefine the reality of the system as experienced by its administrators and users

Thus, Linux security (and UNIX security in general) is a game of “root takes all”

Security in an Open-Source World

- **Open source**

- Software developed by teams that provide access to the source code and all the programming language text from which the final executable is generated
- Source code is open for anyone to see

- **Closed source**

- Commercial software developed by companies that ask you to pay for the program

Security in an Open-Source World

- Most open-source code is free and can be used without restriction.
- Access to open source enables users to learn from source code, see what it does, and understand how a program operates.
- Open source provides the ability for anyone to pick up source code and fix it.
 - Enables speedy resolutions
- Most open-source software is licensed in a way that requires changes to remain open as well.

Security in an Open-Source World

- Open-source code may not be written securely.
- Testing standards vary, and source code may not have sufficient security or regression testing.
- Open-source projects are typically not process heavy.
- If open-source code is modified by a user and then shared, there's no guarantee of a quality fix.

Linux Distributions

- **RedHat**
 - Red Hat Enterprise Linux (RHEL)
 - Fedora Core
- **CentOS** was based on released versions of RHEL, now developed as CentOS Stream (non-Red Hat)
- **Debian**
 - Created by Ian Murdock years ago
 - Debian is a merging of the name Ian with the name Debra (Ian's girlfriend at the time)
 - Mint and Ubuntu are derivatives

Linux Distributions

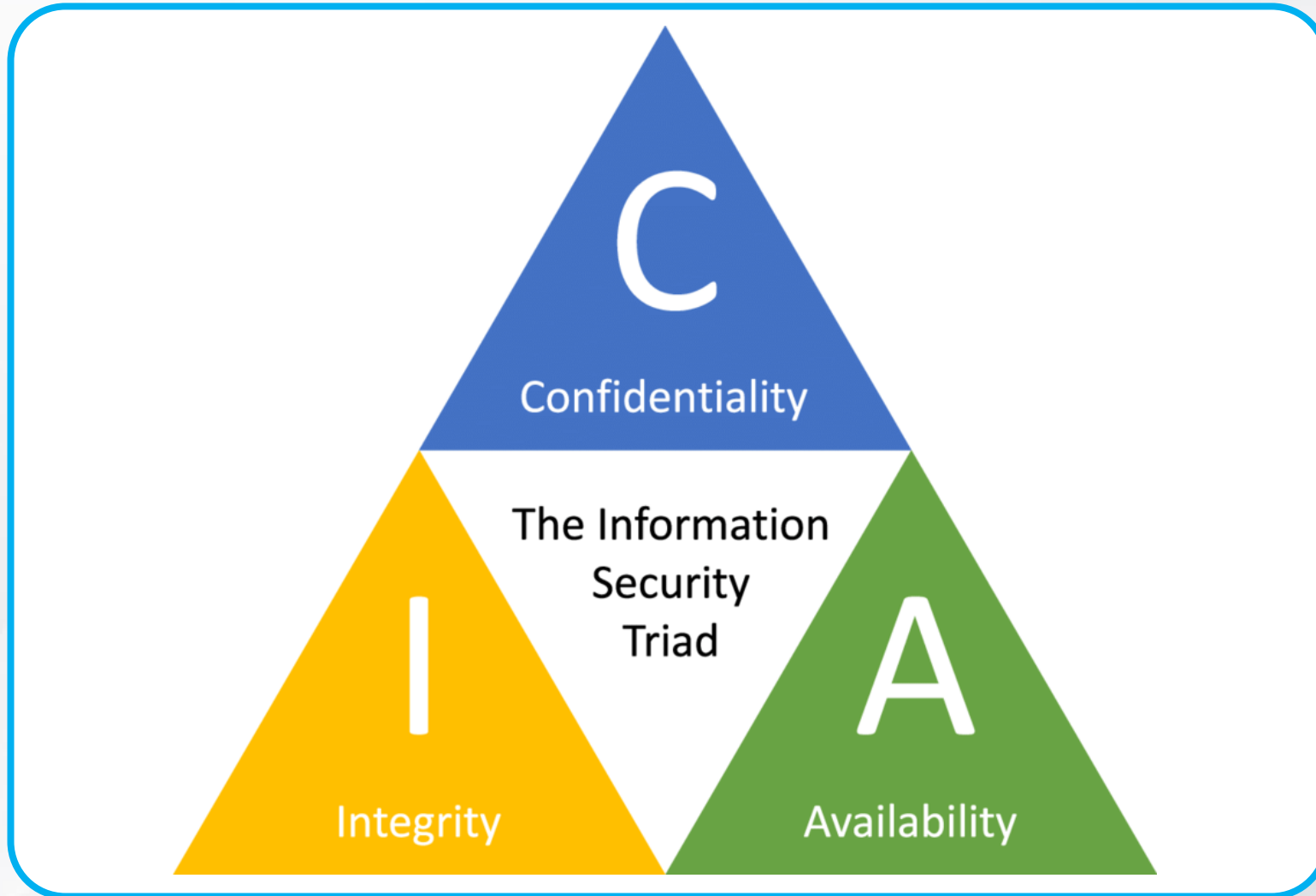
- **Advantages**

- Stable
- Distributions have pre-compiled packages
- Can avoid unnecessary packages by building your own Linux

- **Disadvantages**

- Distribution determines dependencies; may get packages you don't want or need
- Source-based distributions must be compiled from source each time, can be time-consuming

The C-I-A Triad



The C-I-A Triad

- **Confidentiality**
 - Keeping secrets
- **Integrity**
 - Ensuring that the data that are sent are the data that are received
- **Availability**
 - Ensuring use of a service, such as a networked application
 - Software that is buggy and crashes a lot affects availability
 - A denial of service (DoS) attack is a specific and malicious form of a denial-of-service condition

The Parkerian Hexad

- In the late 1990s, Donn Parker, a security professional, proposed an expansion of the C-I-A triad to cover additional concepts he felt were critical to the realm of information security:
 - Confidentiality
 - Possession or control
 - Integrity
 - Authenticity
 - Availability
 - Utility

Linux as a Security Device

- Operating system hardening:
 - Limit the number of software packages installed.
 - Remove all but the most critical users from the system.
 - Restrict permissions on files and directories.
 - Remove all but necessary system services.
- Can deploy a completely hardened operating system as a bastion host
 - A **bastion host** is a system that is presented to the outside world that can be used to protect other systems that are more sensitive.

Linux as a Security Device

- **Firewall**
 - Hardware or software capable of blocking networking communications based on established criteria, or rules
 - Common firewalls available for Linux are iptables and firewalled
- You can set up firewall rules to allow, drop, or log specific packets from the network.
- If there are multiple network interfaces and a network behind a Linux firewall, you can have it perform Network Address Translation (NAT), effectively hiding all the systems behind your Linux firewall.
- On top of the firewall, you can deploy encryption services using Transport Layer Security (TLS).

Linux in the Enterprise

- **Web server**
 - The way customers get access to information, products, and services offered by businesses.
- **Application server**
 - Software that provides a framework inside which applications can be written
- **Security systems**
 - Firewalls and intrusion-detection systems
- **Operating systems**
 - Linux is operating system for many security-based appliances
 - You can use Linux as Proxy server or gateway, jump host, or bastion server

Linux in the Enterprise

- Filtering
 - Between network segments
- Remote Desktop Protocol (RDP) system
 - Many packages support functions commonly found on Windows systems
- File server / file sharing
 - Makes use of Server Message Block (SMB) or Common Internet File System (CIFS) systems, or Network File System (NFS)

Recent Security Issues

- Vulnerabilities
 - Addition of more functions to a system increases complexity
 - Increased complexity results in more potential for unforeseen problems
 - Testing is essential but cannot account for all user behaviors
- Increased prevalence of notification laws
 - Businesses must notify customers of a breach
 - Customers have the right to know when their personal information is at risk

Recent Security Issues

- **White-hat hacker**
 - May use some of the same techniques as a black-hat hacker but is a good guy
- **Gray-hat hacker**
 - Performs a range of actions that a white-hat hacker will not undertake; somewhere in between a white-hat hacker and a black-hat hacker

Recent Security Issues

- The Internet is filled with professional organizations and their skilled employees, running businesses that exist to steal information, extort money, or perform other illegal actions.
- They may be related to the military for different countries, sometimes called nation-states.
- Some attackers look for information, such as intellectual property, that could be sold. They also might engage in corporate espionage or inter-country espionage.

Recent Security Issues

- A malware attack may be launched to get more zombie hosts for a botnet.
 - This is also an attack for financial gain because the zombies may be running web server software as a front to an illegal storefront for, say, pharmaceuticals.
 - The botnet may also be rented to an attacker for political or financial gain.

Summary

- Who are the attackers
- What are the vulnerabilities
- Security features in Operating System
- Operating Systems Hardening
- Security Maintenance
- The origins of Linux
- Open-source software security considerations
- The confidentiality, integrity, and availability (C-I-A) triad
- Linux distributions, Linux security and security threats



Thank You