

# Information Security

Prepared by Dr. Ahmed Omar

IT373  
Information Security

## Lecture 6



# Operating System Security

## Lecture 6

IT373  
Information Security

Prepared by  
**Dr. Ahmed Omar**

## **Agenda**

- ❑ Introduction to Operating System Security
- ❑ Core Security Principles and Mechanisms
- ❑ The Role of the OS Kernel in Security
- ❑ Common OS Vulnerabilities & Attack Vectors
- ❑ OS-Specific Security Analysis

# Introduction to OS Security

## ❑ What is Operating System Security?

The practice of ensuring the **confidentiality**, **integrity**, and **availability** of operating system resources and data



OPERATING SYSTEM SECURITY

## Introduction to OS Security

### □ Why OS Security Matters?

The Operating System is the **central** nervous system of a computer, managing all hardware and software resources.

- If the OS is **compromised**, everything running on it is **compromised**
- OS serves as the **first line** of defense against attacks
- OS enforces all security policies on the machine
- OS mediates access to hardware resources
- A secure OS provides the foundation upon which all other security measures are built.

## Core Goals of OS Security

### ❖ Confidentiality

Preventing unauthorized disclosure of information

**Example:** Ensuring a user cannot read another user's private files.

### ❖ Integrity

Preventing unauthorized modification of data

**Example:** Protecting system files from being altered by malware.

### ❖ Availability

Ensuring that the system and its resources are accessible when needed

**Example:** Preventing Denial-of-Service (DoS) attacks

## Core Security Principles and Mechanisms

To achieve its security goals, the OS employs several fundamental principles and mechanisms.

### 1- Protection vs. Security

- ❑ Protection refers to the internal mechanisms that control the access of programs and processes to system resources.
- ❑ Security, on the other hand, deals with threats from outside the system, such as an unauthorized user trying to gain access.

## Core Security Principles and Mechanisms

### 2- The Principle of Least Privilege

This is one of the most critical concepts in security design. It dictates that a user or process should only be given the minimum levels of access – or permissions – necessary to perform its job functions. For instance, a web server should run as a non-root user to limit the damage if it's compromised.

## Core Security Principles and Mechanisms

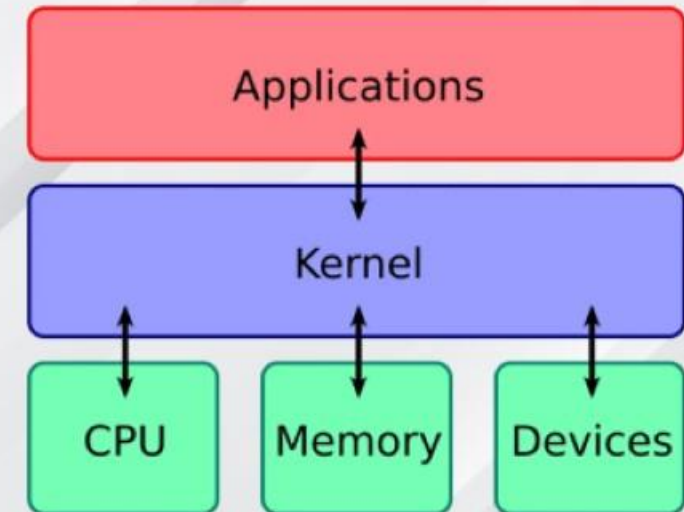
### 3- User Authentication

Before granting access, the OS must verify the identity of a user. The most common form of authentication is the use of **passwords**. However, modern systems increasingly rely on stronger methods like:

- **Multi-Factor Authentication (MFA):** Requiring two or more verification methods.
- **Biometrics:** Using unique physical characteristics like fingerprints or facial recognition.

## OS Kernel Fundamentals

- ❑ The kernel is indeed the ***central part*** of an operating system. It acts as a bridge between applications and the actual data processing done at the hardware level.
- ❑ Because it operates with the highest privileges, the kernel is carefully designed for efficiency and security.



## OS Kernel Fundamentals

The kernel manages critical tasks such as:

- ***Process management:*** deciding which processes run, when, and for how long.
- ***Memory management:*** controlling how memory is allocated and ensuring processes don't interfere with each other.
- ***Device management:*** handling communication between the system and hardware devices like disks, network cards, and keyboards.
- ***System calls and security:*** providing secure ways for applications to interact with the hardware through well-defined interfaces.

## OS Kernel Fundamentals

### Security Functions of the Kernel

- ❑ **Privilege Management:** Enforces separation between privileged (kernel mode) and unprivileged (user mode) operations.
- ❑ **Access Control:** Mediates access to system resources based on permissions and policies.
- ❑ **Memory Protection:** Prevents processes from accessing memory outside their allocated space
- ❑ **System Call Validation:** Verifies the legitimacy and safety of requests from applications

## Protection Rings

- ❑ **Ring 0 (Kernel Mode):** Highest privilege level, direct hardware access.
- ❑ **Ring 3 (User Mode):** Limited privileges, requires system calls to access hardware.



## User Mode vs. Kernel Mode

| Feature          | User Mode              | Kernel Mode             |
|------------------|------------------------|-------------------------|
| Privileges       | Limited                | Full/Unrestricted       |
| Memory Access    | Own virtual space only | All memory areas        |
| CPU Instructions | Restricted set         | All instructions        |
| Hardware Access  | Via system calls only  | Direct access           |
| Crash Impact     | Single process fails   | Entire system may crash |

## How OS is Hacked: Attack Vectors

An **attack vector** is a path or means by which an attacker can gain unauthorized access to a computer or network to deliver a payload or malicious outcome

### Common OS Attack Vectors

- ❑ **Privilege Escalation:** Exploiting vulnerabilities to gain higher access rights
- ❑ **Buffer Overflows:** Overwriting memory boundaries to execute arbitrary code
- ❑ **Command Injection:** Inserting OS commands into application inputs
- ❑ **Zero-Day Exploits:** Attacking unknown vulnerabilities before patches exist

## OS Backdoors: Understanding Covert Access

A backdoor is a typically covert method of bypassing normal authentication or encryption in a computer system, product, or embedded device, providing unauthorized access to data and functions.

- ❑ **Hardware Backdoors:** Implemented at the physical component level, extremely difficult to detect
- ❑ **Software Backdoors:** Implemented in applications or the OS itself, may be hidden in legitimate updates

# OS Backdoors: Understanding Covert Access

## Common OS Backdoor Techniques

- **Hidden User Accounts:** Privileged accounts invisible to normal administration tools.
- **Rootkits:** Malware that provides persistent privileged access while hiding its presence.
- **Authentication Bypasses:** Hardcoded credentials or secret access methods.
- **Covert Channels:** Hidden communication methods for data exfiltration

# System Integrity

**System integrity** refers to the quality of a system performing its intended function in an unimpaired manner, free from unauthorized manipulation, whether intentional or accidental.

## **Key Aspects of System Integrity**

- ***Protection of System Files.*** Preventing unauthorized modification or deletion of critical operating system files and directories.
- ***Process Protection.*** Safeguarding running processes from code injection, runtime manipulation, and other forms of interference
- ***Configuration Integrity.*** Ensuring that system configurations remain as intended and are not tampered with
- ***Data Integrity.*** Maintaining the accuracy and consistency of data throughout its lifecycle

# Secure Boot

A security standard developed by *PC industry members* to ensure a device boots using only software trusted by the *Original Equipment Manufacturer (OEM)*

## How Secure Boot Works

1. Firmware checks digital signatures of each piece of boot software
2. Signature databases (db) contain trusted signatures; revoked signatures database (dbx) contains untrusted ones
3. If signatures are valid, the PC boots and firmware gives control to the operating system
4. If firmware or boot software is untrusted, the boot process is halted

# Linux Security: An Overview

## The Linux Security Philosophy

- ❑ **Open Source:** Transparency allows for community security review
- ❑ **Least Privilege:** Users and processes have minimal permissions needed
- ❑ **Everything is a File:** Unified security model for all resources
- ❑ **Modular Design:** Security components can be added or modified
- ❑ **Multi-User from the Ground Up:** Built with shared systems in mind

# Linux Security: An Overview

## Core Security Strengths

### ❑ Granular permission controls

Linux implements permissions at a highly detailed level, allowing administrators to define what files, commands, and resources each user or process can access.

### ❑ Strong separation between users

Each user account is strictly separated, so compromise of a standard user account does not jeopardize the system as a whole or affect other users

### ❑ Robust logging and auditing capabilities

Linux logs a wide range of system activity, including login attempts, file access, process execution, and administrative actions

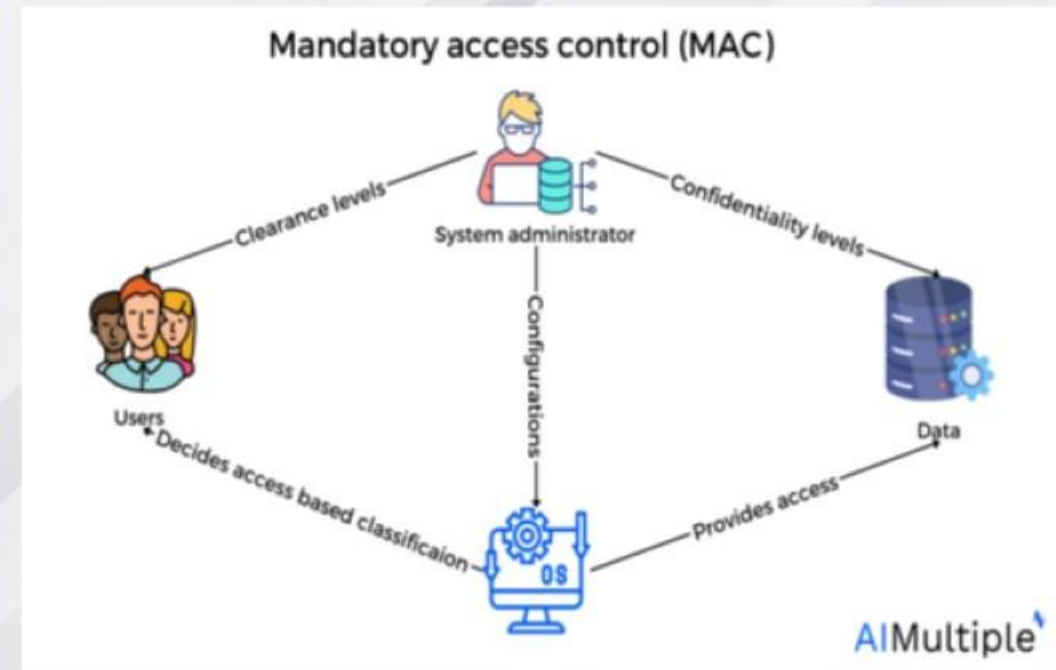
### ❑ Frequent security updates

Linux distributions benefit from large, highly active development communities and vendors who constantly monitor, identify, and fix vulnerabilities

# Linux Security Features

## Mandatory Access Control (MAC)

- MAC is a security model where access to resources is controlled by a system-wide policy enforced by the OS kernel.
- MAC provides a much higher level of security, often used in government and enterprise environments.



# Linux Security Features

## Linux kernel security modules

**SELinux** (Security-Enhanced Linux) and **AppArmor** (Application Armor) are both Linux kernel security **modules** that provide a layer of security to Linux systems. They both implement Mandatory Access Control (**MAC**), which allows administrators to set a centralized security policy.

This is in contrast to the standard **Discretionary Access Control** (DAC) in Linux, where users and applications can access the entire system if they have the correct permissions.

# Linux Security Features

## Linux kernel security modules

### ❑ **SELinux:**

***Security-Enhanced Linux*** provides fine-grained control over access to resources beyond traditional permissions.

### ❑ **AppArmor:**

Application profiles that restrict programs' capabilities and access to files

**AppArmor** is generally considered to have a *lower learning curve* and is easier to manage than **SELinux**.

# Linux Security Features

| Feature        | SELinux   | AppArmor |
|----------------|-----------|----------|
| Complexity     | Higher    | Lower    |
| Granularity    | Very Fine | Moderate |
| Learning Curve | Steep     | Moderate |
| Security Level | Very High | High     |

# Linux Security Features

## User and Group Permissions

### ❑ Traditional Unix Permissions:

Read, Write, Execute permissions for Owner, Group, and Others

### ❑ Principle of Least Privilege:

Users and processes operate with minimal permissions needed

## Access Control Lists (ACLs)

- Extends traditional permissions with more fine-grained control
- Allows specifying permissions for multiple users and groups on a single file

# Linux Security Features

## ☐ **Firewalls (iptables, nftables, ufw)**

Network traffic control tools that filter packets based on rules, protecting against unauthorized network access and attacks.

## ☐ **UEFI Secure Boot**

Ensures only digitally signed bootloaders and kernels can be loaded during startup, preventing boot-time malware.

## ☐ **Linux Kernel Lockdown**

Restricts access to kernel features even for the root user, enhancing system integrity by preventing kernel modifications.

## ☐ **Logging and Monitoring**

Tools like journalctl, logrotate, and AIDE (Advanced Intrusion Detection Environment) for tracking system events and detecting unauthorized changes.

## ☐ **Fail2ban**

Monitors logs for suspicious activity (e.g., repeated failed login attempts) and automatically blocks offending IP addresses.

# Linux Security Features

## Common Vulnerabilities & Attacks

- ❑ **Privilege Escalation:** E.g., OverlayFS vulnerabilities (CVE-2023-0386)
- ❑ **Out-of-Bounds Access:** Memory manipulation in kernel drivers
- ❑ **Weak Credentials:** SSH brute-force attacks targeting default or weak passwords
- ❑ **Web Application Attacks:** SQL injection and code injection vulnerabilities

## Key Countermeasures

- ✓ **Regular Updates:** Keep kernel and packages patched
- ✓ **Principle of Least Privilege:** Run services with minimal permissions
- ✓ **Firewall Configuration:** Implement strict firewall rules
- ✓ **Intrusion Detection:** Deploy IDS/IPS to monitor suspicious activity

# Windows Security Features

## The Windows Security Philosophy

- ❑ **Integrated Protection:** Security features built directly into the OS
- ❑ **User-Friendly:** Security tools designed for ease of use
- ❑ **Defense in Depth:** Multiple layers of security working together
- ❑ **Centralized Management:** Security features managed through Windows Security



# Key Security Features

## ❑ Windows Security App

Comprehensive security solution integrated into Windows, providing a central hub for managing various security settings

## ❑ Microsoft Defender Antivirus

Built-in real-time protection against viruses, malware, spyware, and other threats with continuous scanning capabilities.

## ❑ Windows Firewall

Controls network traffic by monitoring incoming and outgoing connections, blocking suspicious activity based on predefined rules.

## ❑ Secure Boot

Ensures only digitally signed and trusted bootloaders and OS components are loaded during startup, preventing boot-time malware from compromising the system.

# Key Security Features

## ☐ **User Account Control (UAC)**

Prevents unauthorized changes to the system by prompting for permission before allowing programs to make significant modifications

## ☐ **BitLocker Drive Encryption**

Full disk encryption that protects data on the entire drive, making it unreadable if the device is lost or stolen. Particularly important for laptops and portable devices.

## ☐ **Windows Hello**

provides secure biometric and PIN authentication to control user access

These features work together to provide a layered security approach, protecting against various types of threats

# Windows Security Features

## Common Vulnerabilities & Attacks

- ❑ *Remote Code Execution (RCE)*: E.g., SPNEGO Extended Negotiation vulnerability (CVE-2025-47981)
- ❑ *Elevation of Privilege*: Exploits in Windows components like CLFS
- ❑ *Malware & Ransomware*: Most prevalent threats targeting Windows
- ❑ *Phishing & Identity-based Attacks*: Targeting user credentials

## Key Countermeasures

- **Regular Updates**: Apply Patch Tuesday updates promptly
- **Strong Authentication**: Implement MFA and complex passwords
- **User Education**: Train users to recognize threats

# Windows Security Features

## The Importance of Updates

Windows Update is perhaps the single most important security feature in Windows, providing a mechanism for Microsoft to deliver

- ✓ **Security Patches**

Fixes for vulnerabilities that could be exploited by attackers

- ✓ **Feature Updates**

New functionality and improvements to existing features

- ✓ **Definition Updates**

New malware signatures for Windows Defender



# Android Security Features

## Key Security Features

- ❑ ***Application Sandbox:*** Isolates apps from each other and the system
- ❑ ***App Signing:*** Ensures apps come from verified developers
- ❑ ***Encryption:*** Full-disk and file-based encryption for data protection
- ❑ ***Google Play Protect:*** Built-in malware protection service



# Android Security Features

## Common Vulnerabilities & Attacks

- ❑ **Remote Code Execution:** Critical vulnerabilities in system components
- ❑ **Untrusted APKs:** Malicious apps from unofficial sources
- ❑ **Phishing:** SMS and email-based attacks to steal credentials
- ❑ **Spyware:** Apps that secretly monitor user activities

## Key Countermeasures

- **Regular Updates:** Install security patches promptly
- **Official App Sources:** Only download from Google Play Store
- **Permission Review:** Carefully review app permissions
- **Avoid Rooting:** Maintain built-in security mechanisms

# macOS Security Features

## Key Security Features

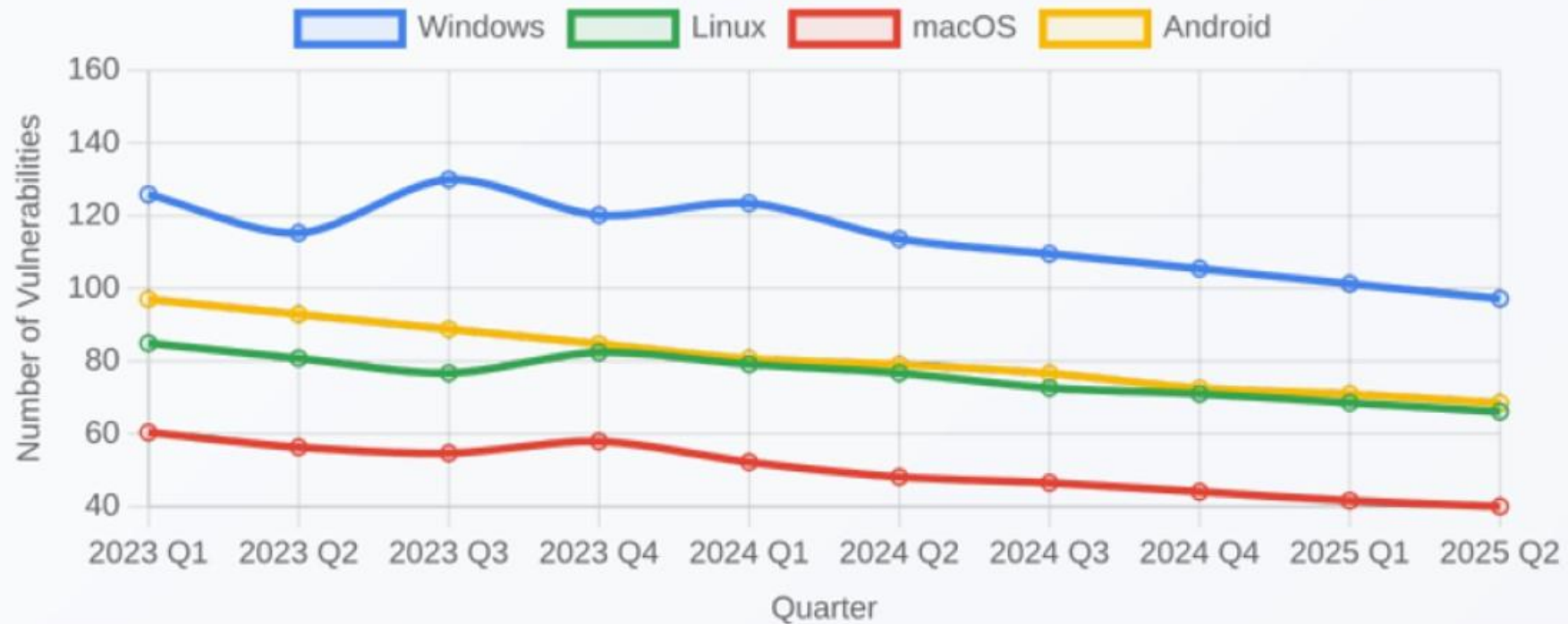
- 🛡️ **System Integrity Protection (SIP):** Protects critical system files from modification
- 📁 **App Sandboxing:** Isolates applications to limit access to system resources
- 🔒 **FileVault Encryption:** Full-disk encryption using XTS-AES 128
- ✅ **Gatekeeper:** Ensures only trusted software runs on the system

## Common Vulnerabilities & Attacks

- 🚫 **SIP Bypass:** Vulnerabilities like CVE-2024-44243 that bypass system protection
- ⚠️ **Zero-day Exploits:** Actively exploited vulnerabilities before patches
- 🦋 **Malware:** Including adware, info-stealers, and cryptocurrency miners
- 🏠 **Phishing:** Sophisticated social engineering targeting Apple users

# OS Security Comparison

## Vulnerability Trends (2023-2025)



# OS Security Comparison

## Security Considerations

- ❑ **Attack Surface:** Windows and Android have larger attack surfaces due to market share and diversity.
- ❑ **Default Security:** macOS offers strongest out-of-box security, followed by Linux.
- ❑ **Update Frequency:** All major OSes now provide regular security updates

# Key Takeaways

- ❑ No operating system is completely secure; each has strengths and vulnerabilities that must be understood and managed.
- ❑ Security is a continuous process, not a one-time implementation.
- ❑ Regular updates, monitoring, and adaptation are essential.
- ❑ A multi-layered security approach that combines technical controls with user education provides the strongest defense.

# Summary

- ✓ **Fundamentals of OS Security**
- ✓ **Core Security Principles**
- ✓ **The Role of the OS Kernel**
- ✓ **OS-Specific Security Features**



**Thank You**

