Federico Cinelli

CS510 – Operating System Principles

Module 1 – Journal Entry

**Operating System Security and Access Control: A Comparative Analysis of Windows, macOS, and Linux**

Operating systems are fundamental to computing environments, not only for managing hardware and software resources but also for enforcing system security and access control. They serve as the gatekeepers of digital systems, regulating who can access data, execute programs, and interact with hardware components. Through mechanisms such as user authentication, file permissions, and process isolation, operating systems ensure that only authorized users and applications can perform sensitive operations, thereby protecting the integrity and confidentiality of the system.

Windows, macOS, and Linux each implement distinct security architectures tailored to their design philosophies and user bases. Windows uses a layered security model that includes components like the Security Reference Monitor (SRM), Local Security Authority (LSA), and Active Directory for enterprise-level identity management. It uses Discretionary Access Control (DAC) and Mandatory Integrity Control (MIC), with Access Control Lists (ACLs) defining permissions for files and system objects. Authentication in Windows is robust, supporting password-based login, biometric verification through Windows Hello, and smart card integration. Encryption is handled through BitLocker for full-disk protection and the Encrypting File System (EFS) for individual files. Windows maintains system integrity through User Account Control (UAC) and code signing, ensures confidentiality via encryption and access restrictions, and promotes availability through regular updates and recovery tools.

macOS, built on a Unix foundation, emphasizes user-centric security with strong default protections. Its architecture includes System Integrity Protection (SIP), Gatekeeper, and XProtect, which collectively prevent unauthorized modifications and block malicious software. Access control is enforced through POSIX-compliant permissions and application sandboxing. Authentication methods include passwords, biometric verification via Touch ID, and Apple ID-based access. macOS uses FileVault for full-disk encryption and relies on the Secure Enclave for advanced cryptographic operations. Integrity is preserved through SIP and app notarization, confidentiality is maintained through encrypted storage and secure app distribution, and availability is supported by seamless system updates and Time Machine backups.

Linux, known for its modular and open-source nature, offers granular control over security settings. Its architecture supports kernel-level security modules such as SELinux and AppArmor, which enforce Mandatory Access Control (MAC) policies. Linux also uses DAC through traditional file permissions and user roles. Authentication is managed via password-based login, Pluggable Authentication Modules (PAM), and integration with LDAP or Kerberos for centralized identity management. Encryption tools like LUKS and GnuPG provide disk and file-level protection, while SSH keys are widely used for secure remote access. Linux ensures integrity through package signing and audit logs, maintains confidentiality with strict user privileges and encryption, and supports availability through community-driven patching and system redundancy.

Despite their differences, Windows, macOS, and Linux share common goals in upholding the principles of confidentiality, integrity, and availability. Windows prioritizes enterprise integration and user-friendly security tools, macOS focuses on seamless and intuitive security for end users, and Linux offers transparency and flexibility for administrators and power users. Each system’s approach reflects its target audience and design philosophy, but all three demonstrate a commitment to robust and evolving security practices.