# Unit 4: Smartphone Security

## Android vs. iOS Security Model

### Android Security Model

- Foundation: Based on the Linux kernel, Android’s security architecture leverages established Linux mechanisms such as process isolation, file permissions, and Inter-Process Communication (IPC).  
- Key Security Features:  
 - Process Isolation: Each app runs in its own process and has its own unique Linux user ID (UID). This ensures apps cannot access each other's data without explicit permissions.  
 - User-Based Permissions: Apps must request permissions for accessing sensitive data or system features. Permissions are granted at runtime, allowing users to make informed decisions.  
 - Inter-Process Communication (IPC): Secure mechanisms such as Binder facilitate communication between app components while enforcing access controls.  
- Threat Model: Android assumes a diverse threat landscape including malicious apps, physical theft, and network-based attacks. Regular updates and Google Play Protect provide ongoing defenses.

### iOS Security Model

- Architecture:  
 - iOS is structured into four layers: Core OS (kernel and hardware interactions), Core Services (essential system services), Media (graphics and audio frameworks), and Cocoa Touch (user interface and app interaction).  
- Key Features:  
 - Secure Boot Chain: Ensures that only Apple-signed software runs on devices by verifying each step in the boot process, preventing unauthorized code execution.  
 - Sandboxing: Apps operate in isolated environments, restricting their access to system resources and user data unless explicitly permitted.  
 - Mandatory Code Signing: All executable code must be signed by Apple, ensuring app authenticity and integrity.  
- Privacy and Security:  
 - Data encryption, such as AES-256, protects files and personal information.  
 - Features like Face ID and Touch ID enhance security while maintaining user convenience.

## Threat Models

- Android Threats:  
 - Malware: Often introduced through untrusted sources like third-party app stores.  
 - Rooting Vulnerabilities: Exploiting kernel flaws to gain unauthorized privileges.  
 - Repackaging Attacks: Legitimate apps are modified to include malicious code and redistributed.  
- iOS Threats:  
 - Jailbreaking: Weakens device security by bypassing Apple's software restrictions.  
 - Third-Party App Exploits: Vulnerabilities in apps can lead to data leakage or unauthorized access.  
 - Phishing and Social Engineering: Users may be tricked into revealing sensitive information through deceptive messages or emails.

## Information Tracking

- Android: User data can be accessed through permissions granted to apps. Poor permission management or malicious apps may lead to extensive data tracking and leakage.  
- iOS: Apple’s stringent app approval process minimizes unauthorized tracking. Apps must explicitly disclose their data usage policies and seek user consent for tracking.

## Rootkits

- Android:  
 - Rootkits exploit system vulnerabilities to gain persistent, unauthorized access to device resources.  
 - Commonly delivered via malware or malicious apps, targeting system files or kernel processes.  
- iOS:  
 - Rare due to the secure boot chain and code signing requirements.  
 - Potential threats arise primarily on jailbroken devices, which bypass Apple’s security mechanisms.

## Access Control in Android Operating System

- Android employs a Discretionary Access Control (DAC) model:  
 - Each app operates under a unique Linux UID and has dedicated file storage.  
 - Permissions control access to system resources like the camera, microphone, or contacts. These permissions are explicitly requested in the app manifest.  
 - SELinux (Security-Enhanced Linux) enforces mandatory access control policies, enhancing the overall security framework.

## Rooting Android Devices

- Definition: Rooting grants superuser access to the Android operating system, bypassing built-in security measures.  
- Advantages:  
 - Enables deep customization and access to advanced features.  
 - Allows installation of specialized apps requiring root permissions.  
- Risks:  
 - Compromises security by disabling OS protections.  
 - Increases vulnerability to malware and rootkits.  
 - Voids warranties and may render the device unstable.  
- Best Practices:  
 - Root only if absolutely necessary and use trusted tools.  
 - Install robust security apps to monitor for vulnerabilities.

## Repackaging Attacks

- Definition: Malicious actors modify legitimate apps by injecting harmful code and repackaging them.  
- Steps Involved:  
 1. The attacker downloads the legitimate app and decompiles it.  
 2. Malicious payloads are added.  
 3. The app is recompiled and distributed via unofficial channels.  
- Consequences:  
 - Users may unknowingly install the tampered app, compromising their devices.  
 - Sensitive data such as passwords or banking information can be stolen.  
- Preventive Measures:  
 - Always download apps from official app stores.  
 - Use tools to verify app integrity and signatures.

## Attacks on Apps

- Types of Attacks:  
 - Data Theft: Sensitive data is extracted due to weak encryption or insecure APIs.  
 - Injection Attacks: Malicious commands are injected into app inputs to manipulate backend systems.  
 - Phishing Attacks: Users are deceived into providing credentials through fake app interfaces.  
- Countermeasures:  
 - Secure APIs with authentication and input validation.  
 - Use SSL/TLS to encrypt communications.  
 - Regularly update apps to address security vulnerabilities.

## Whole-Disk Encryption

- Android:  
 - Encrypts all user data using hardware-backed AES encryption keys.  
 - Introduced mandatory full-disk encryption in Android 5.0, enhanced with file-based encryption in later versions.  
- iOS:  
 - Employs hardware-based AES-256 encryption to safeguard all data on the device.  
 - Utilizes a multi-tiered key hierarchy to protect data at rest, ensuring only authorized users can access sensitive files.

## Hardware Protection

- Android:  
 - Relies on Trusted Execution Environment (TEE) and TrustZone to isolate sensitive operations.  
 - Hardware-backed Keystore protects cryptographic keys.  
- iOS:  
 - The Secure Enclave is a dedicated coprocessor for handling cryptographic operations and key management.  
 - Features like Touch ID and Face ID utilize Secure Enclave for biometric authentication.

## Viruses, Spyware, and Keyloggers

- Android:  
 - Malicious apps can introduce spyware and keyloggers, especially on rooted devices.  
 - Antivirus apps can help detect and mitigate threats.  
- iOS:  
 - Rigid app vetting and sandboxing minimize risks.  
 - Jailbroken devices are more susceptible to spyware and keyloggers due to disabled security features.

## Malware Detection

- Android:  
 - Regularly update the OS to patch vulnerabilities.  
 - Use security solutions like Google Play Protect and third-party antivirus software.  
- iOS:  
 - Apple’s App Store review process reduces the risk of malware.  
 - Users should avoid jailbreaking and be cautious of phishing attempts.

# Unit 5: Hardware and System Security

## 1. Meltdown Attack

What is Meltdown?

Meltdown is a critical vulnerability in modern CPUs that allows malicious programs to access privileged memory, including sensitive data like passwords and encryption keys. It exploits a flaw in speculative execution, a performance optimization technique used by processors. Discovered in 2018, Meltdown affects Intel processors, some ARM-based chips, and other architectures, making it a significant threat to global cybersecurity.

How it Works:

The CPU speculatively executes instructions that access memory unauthorizedly. Even if these instructions are discarded, the side effects (like changes in the CPU cache) are observable. Attackers use these side effects to infer the contents of protected memory.

Applications:

Data theft from protected kernel memory.

Mitigations:

Kernel Page-Table Isolation (KPTI), microcode updates, and software patches.

## 2. Spectre Attack

What is Spectre?

Spectre is a class of vulnerabilities that trick a CPU into executing instructions speculatively on incorrect assumptions, leading to unauthorized access to sensitive information. Named after 'speculative execution,' Spectre exploits vulnerabilities in the branch prediction mechanism of CPUs. First revealed in 2018, Spectre affects almost all modern processors from Intel, AMD, and ARM.

How it Works:

Attackers craft malicious code that manipulates the CPU's branch predictor. This speculative execution leads to unauthorized memory access. Side-channel techniques, like cache-timing analysis, are then used to infer sensitive data.

Applications:

Gaining access to data across virtual machines or within the same application.

Mitigations:

Software fixes, like Retpoline, and architectural changes in new processors.

## 3. Authentication and Passwords

What is Authentication?

Authentication is the process of verifying the identity of a user, system, or device to ensure secure access to resources. Common authentication methods include passwords (knowledge-based), tokens (possession-based), and biometrics (inherent-based). Multi-factor authentication (MFA) combines these methods for enhanced security.

How it Works:

Passwords are hashed and stored securely. During login, the provided password is hashed and compared to the stored hash. MFA adds layers, like OTPs or biometric scans, to verify identity further.

Applications:

Used in systems ranging from personal devices to enterprise networks.

Mitigations:

Regular password updates, strong password policies, and use of password managers.

## 4. Access Control Concepts

What is Access Control?

Access control determines who can access a resource and what actions they can perform. Two main types:

- Discretionary Access Control (DAC): Users control resource permissions.

- Mandatory Access Control (MAC): Policies strictly define permissions.

Role-Based Access Control (RBAC) is widely used in organizations, assigning permissions based on user roles.

## 5. Access Control List (ACL)

What is an ACL?

An ACL is a list of permissions attached to an object, specifying which users or groups can access it and what operations they can perform. Examples include file systems and network firewalls.

How it Works:

The ACL contains entries defining access levels for users/groups. The system evaluates these entries when access is requested.

Applications:

Restricting access to sensitive files or controlling network traffic.

## 6. Capability

What is a Capability?

A capability is a token that grants access rights to a resource. Unlike ACLs, capabilities are distributed and act as keys.

How it Works:

Capabilities are securely issued by the system. When access is requested, the capability is validated, ensuring permission adherence.

Applications:

Used in distributed systems, API authentication, and secure delegation of rights.

## 8. Threats of Hardware Trojans and Supply Chain Security

What are Hardware Trojans?

Hardware Trojans are malicious modifications to a hardware design, often introduced during manufacturing.

How They Work:

Trojans activate under specific conditions to compromise functionality, leak data, or create backdoors.

What is Supply Chain Security?

Ensuring the integrity of components throughout their lifecycle, from design to deployment.

How it Works:

Audits, secure transport, and tamper-proof measures are employed to safeguard the supply chain.

Applications:

Critical infrastructure, consumer electronics, and IoT devices.

## 9. Side-Channel Analysis-Based Threats and Attacks

What are Side-Channel Attacks?

Side-channel attacks exploit physical characteristics of systems (e.g., timing, power consumption) to extract sensitive data.

How They Work:

Example: Observing power usage during cryptographic operations to infer keys.

Applications:

Attacking smart cards, secure processors, and IoT devices.

Mitigations:

Design hardware with noise and randomization to prevent leakage.

## 10. Issues in Critical Infrastructure and SCADA Security

What is SCADA?

* SCADA (Supervisory Control and Data Acquisition) is a combination of software and hardware components that work together to monitor and control industrial processes. It is commonly used in power generation, water treatment, manufacturing, and other large-scale industrial applications.
* SCADA systems provide high-level supervision and data acquisition, enabling operators to monitor the status of devices and processes, detect anomalies, and make informed decisions.
* SCADA consists of components like Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and Human-Machine Interfaces (HMIs), each playing a key role in the system's functionality.

How it Works:

* Real-time operation: Controls physical processes requiring immediate attention.
* Geographic distribution: Often spans large areas, making security management challenging.
* Legacy systems: Many are based on outdated technology, not designed with modern security.
* Limited resources: Often constrained by processing power and memory.

**Security Threats:**

* Cyberattacks: Malware, ransomware, and phishing can target SCADA systems, leading to data corruption, system shutdowns, or theft of sensitive information.
* Unauthorized Access: Attackers gaining unauthorized access can manipulate or disrupt SCADA processes, potentially affecting operations and safety.
* Data Breaches: SCADA systems often store critical data. If compromised, it can lead to exposure of sensitive or proprietary information.
* Physical Attacks: Physical sabotage or vandalism can disrupt the functioning of SCADA infrastructure, such as damage to servers or communication equipment.
* Insider Threats: Employees or contractors with malicious intent or lack of awareness can cause security breaches or unintended disruptions to the system.

**Mitigations:**

* Implement perimeter protection measures like firewalls, intrusion detection systems, and VPNs.
* Use interior security practices, including access control, host-based protections, and network segmentation.
* Regularly audit and test SCADA systems for vulnerabilities.
* Ensure robust monitoring and incident management systems are in place.
* Secure remote devices like RTUs and PLCs by limiting access and employing encrypted communication protocols.