CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

INSTITUTE : UIE

DEPARTMENT : CSE

Bachelor of Engineering (Computer Science & Engineering)

WEB AND MOBILE SECURITY (Professional Elective-I)

(20CST/IT-333)

TOPIC OF PRESENTATION:

IOS, Android and Window phone Security Model

Lecture Objectives

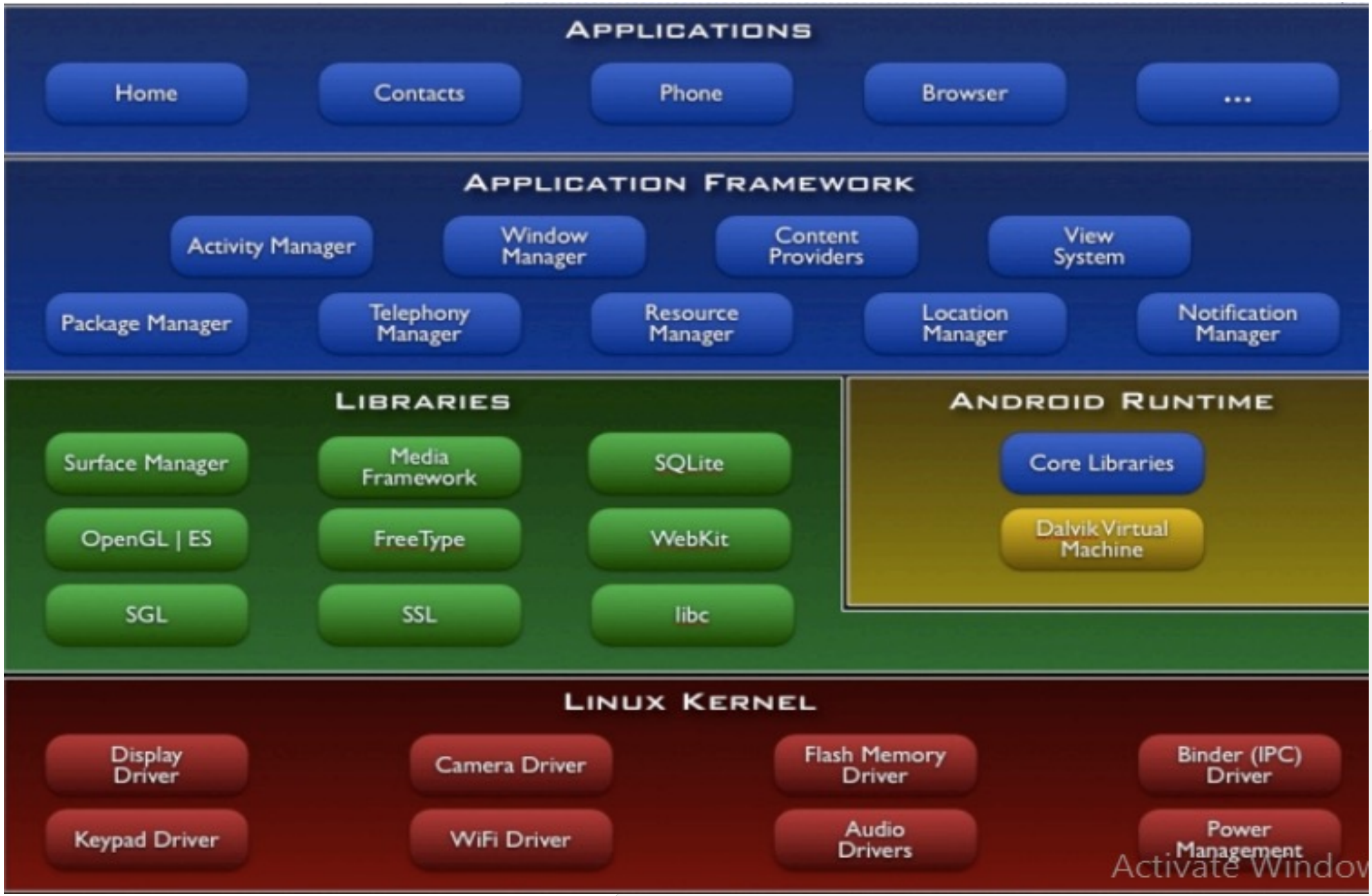
In this lecture, we will discuss:
IOS, Android and Window phone Security Model

Introduction

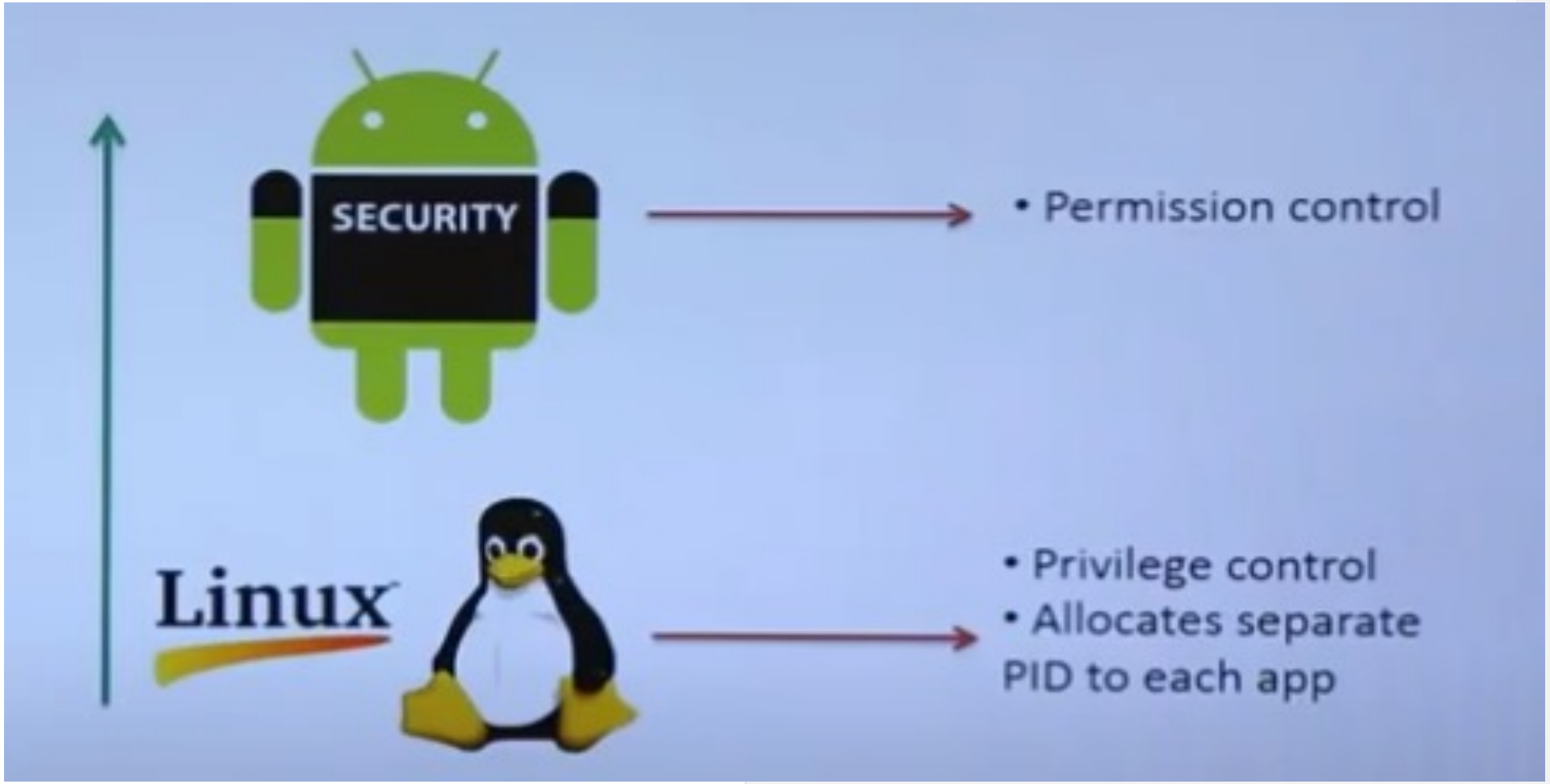
- Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services.
- Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and **Dalvik Virtual Machine (DVM)** provide platform for running an android application.

The main components of android architecture are following:-

- Applications
- Application Framework
- Android Runtime
- Platform Libraries
- Linux Kernel



Android security Model



The Layers of the Android Security Model

The security model is based on the consent of the following parties:

1. **Operating System**
2. **Application**
3. **End-User**

1. Operating System

The kernel security determines the overall security of the whole system. The security of the Android operating system is based around the following key security features of the Linux kernel:

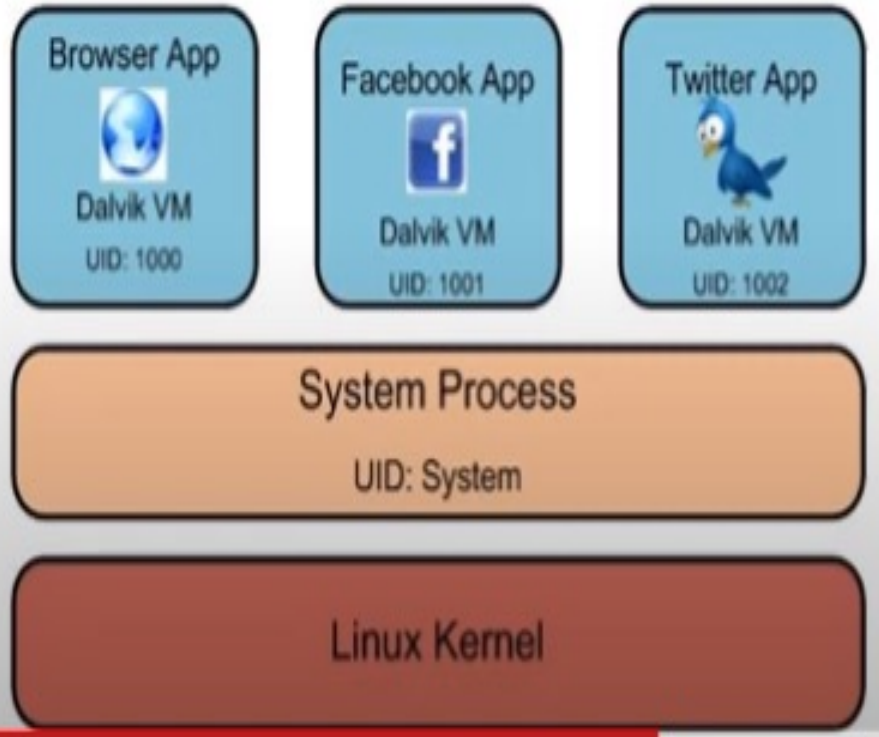
- **Process Isolation(sandboxing)**
- **User-Based Permission Model**
- **Inter-Process Communication (IPC)**

- The secure communication between applications is ensured by the Linux *user-based protection*.
- Android platform uses the Linux *user-based permissions model* to isolate application resources. This process in called **application sandbox**.
- The aim of sandboxing is to **prevent malicious external programs from interacting with the protected app**.
- The internal operating system components are also protected by the sandboxing mechanism.
- Unlike traditional operating systems, e.g. MacOS and Windows, Android uses the **User ID (UID)** concept to manage an **application's access control** and not the system user's access control.
- An application is prohibited from accessing other application's data or system features without the necessary permissions.
- The application is sandboxed at the *kernel level*, hence it is guaranteed that the application is isolated from the rest of the system, regardless of specific development environment, programming languages or APIs used.

The Android security model is **primarily based on a permission mechanism**.

Each application is running in a specific Dalvik virtual machine with a unique user ID assigned to it, which means the application code runs in isolation from the code of all others applications.

As a result, one application has not granted access to other applications' files.



Rooting

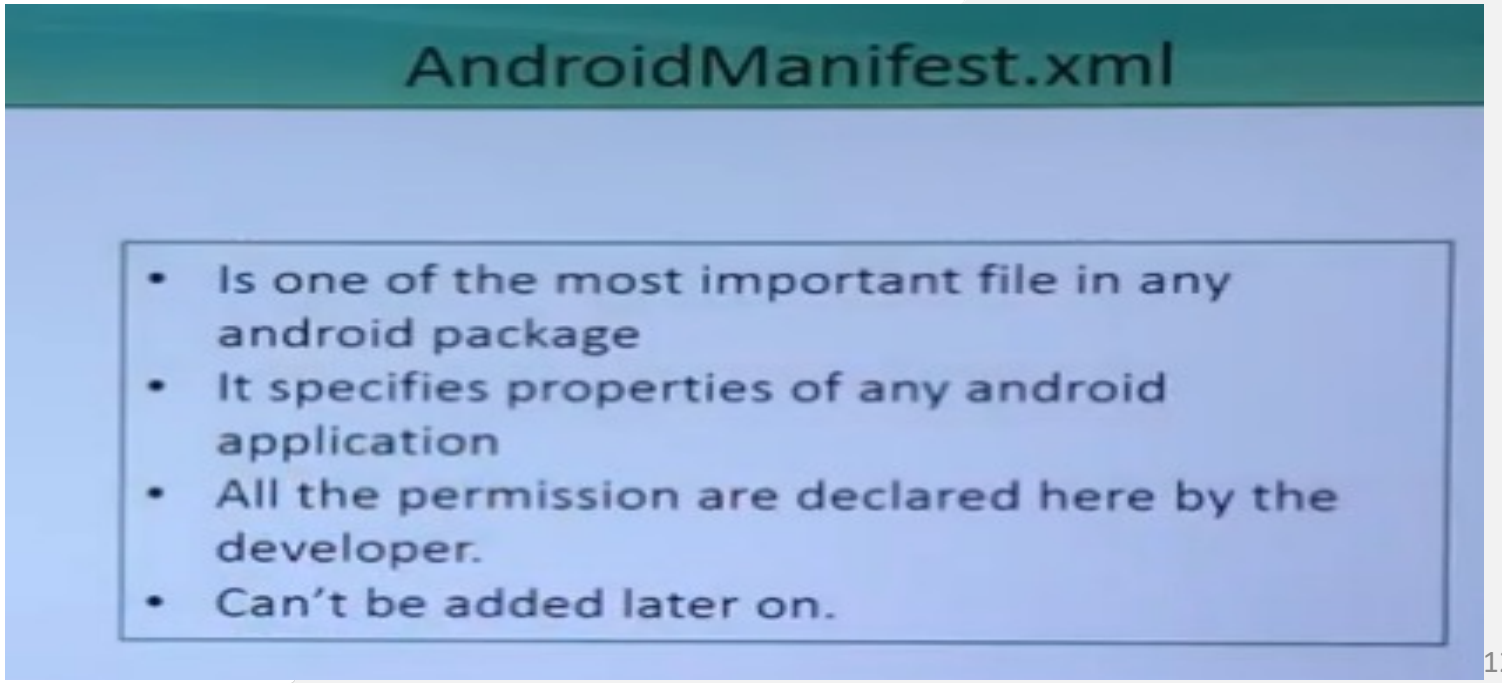
- On a Linux system, *root* is the name of the **account that has access to all files and commands**. Rooting is the process of gaining access to more administrative-level controls on an Android device. Despite its benefits, attackers often use rooting to target sensitive user and business data.
- Owner of the device is **NOT** root. Android is designed to be **open**. Consequently, the user is allowed to root the phone, i.e. switch to the *root* user.
- It should be noted that rooting is not recommended to be done by inexperienced users. Rooting a phone might result in making the warranty null.
- Verified Boot is a process that ensures that the device is booting the *original operating system* alongside the afferent *system code*, and not a malicious replica. It establishes a *chain of trust* between the multiple components that can be altered, starting from the hardware up to the verified partitions

- Without rooting, users cannot access or modify system files and folders. Once rooted, the user has full access to the device. Rooting allows the user to make changes to everything on the device. This allows users to do things that were previously impossible, like removing bloatware, customization, custom ROMs, etc.
- Rooted devices may contain many apps that process sensitive information, such as banking apps, payment apps, social media, and cloud storage. Malicious downloads can expose your device to hackers.

https://www.appknox.com/blog/root-detection-techniques

2. Application Security

- **Permissions**
- In Android, the user's privacy is protected by the means of **permissions.** Android applications requires the user's consent to perform actions. The manifest file **describes essential information about your app to the Android build tools, the Android operating system, and Google Play**.
- The permissions required by an application are declared in the ***AndroidManifest.xml***. Every permission is specified in its own ***uses-permission*** tag.



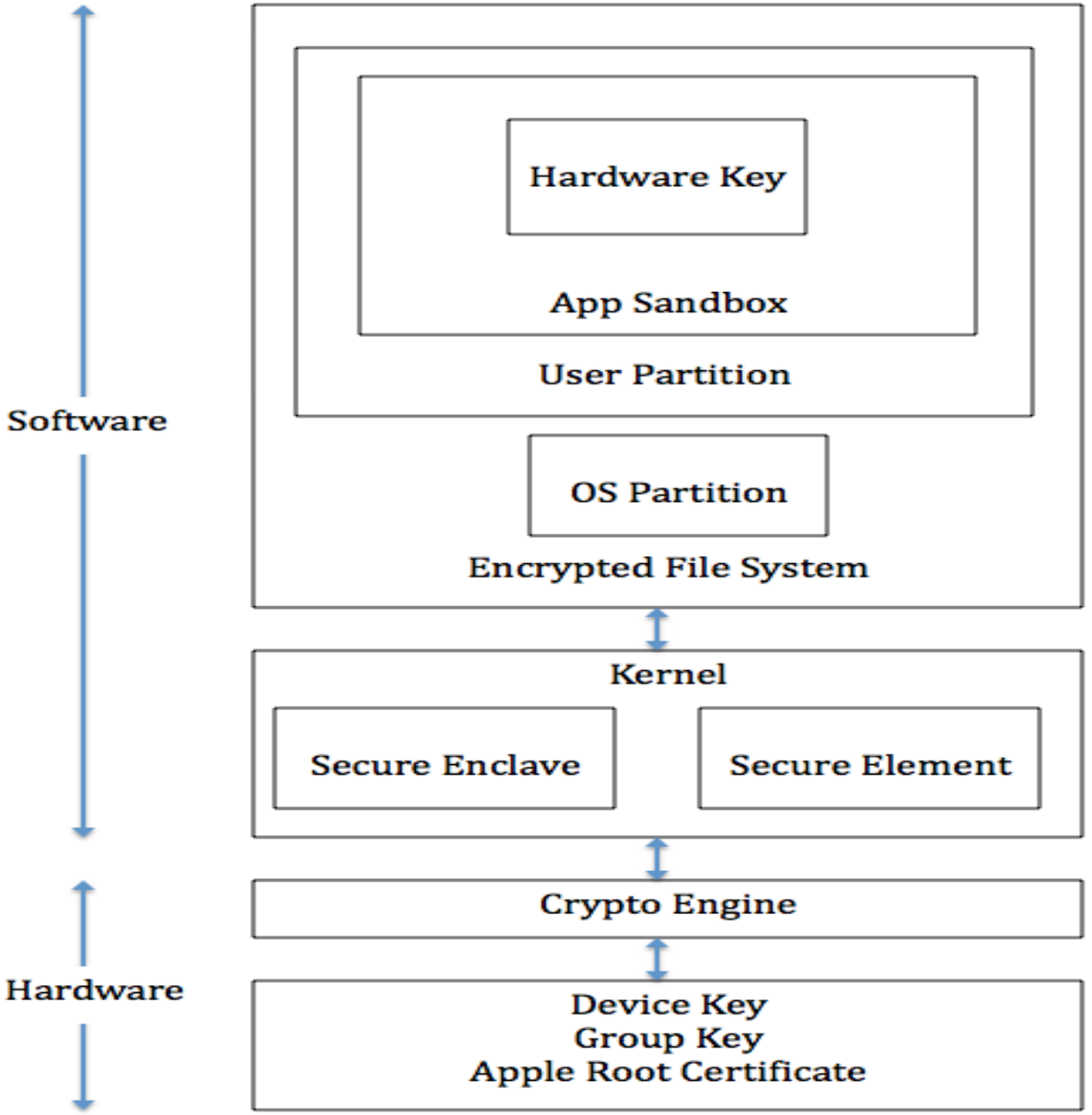
Application Signing

- Application signing is the first step in the application sandbox mechanism.
- **An UID is assigned based on the certificate used to sign the application**.
- Code signing provides developers with the ability to **identify the author of an application**
- Applications that are not signed by a developer cannot be uploaded to Google Play.
- Even if an unsigned application package ended up on the device, the application cannot be installed because the package manager checks whether the package is signed or not before installing any application.

Application sandbox

- Application sandbox
  - Each application runs with its UID in its own Dalvik virtual machine
    - Provides CPU protection, memory protection
    - Authenticated communication protection using Unix domain sockets
    - Only ping, zygote (spawn another process) run as root
  - Applications announces permission requirement
    - Create a whitelist model – user grants access
      - But don't want to ask user often – all questions asked as install time
    - Inter-component communication reference monitor checks permissions

IOS Security Model

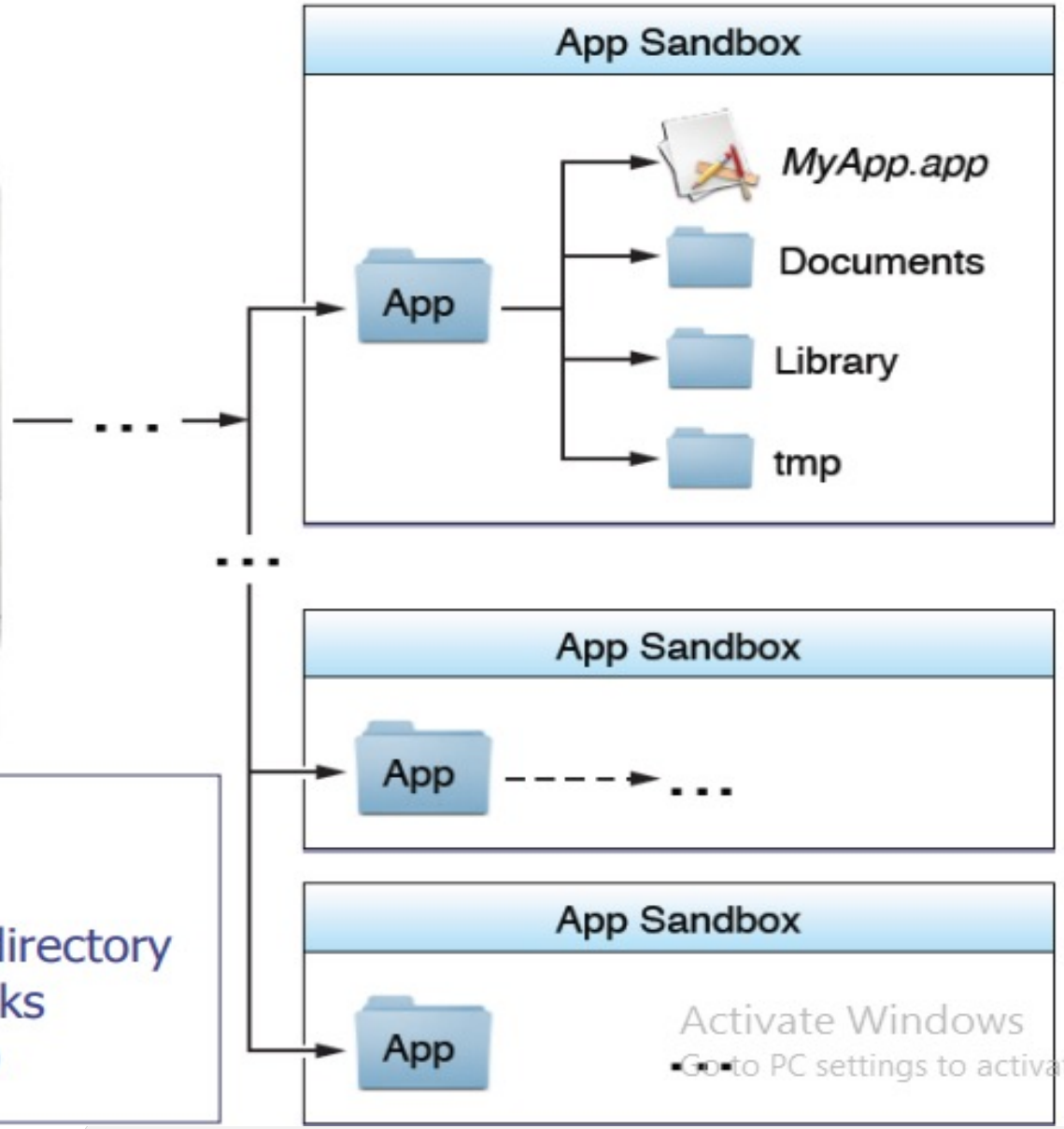


Apple iOS Security

- Device security
  - Prevent unauthorized use of the device
- Data security
  - Protect data at rest; device may be lost or stolen
- Network security
  - Networking protocols and encryption of data in transmission
- App security
  - Secure platform foundation

https://crypto.stanford.edu/cs155old/cs155-spring14/lectures/17-mobile-platforms.pdf

iOS Sandbox



- Limit app's access to files, preferences, network, other resources
- Each app has own sandbox directory
- Limits consequences of attacks
- Same privileges for each app

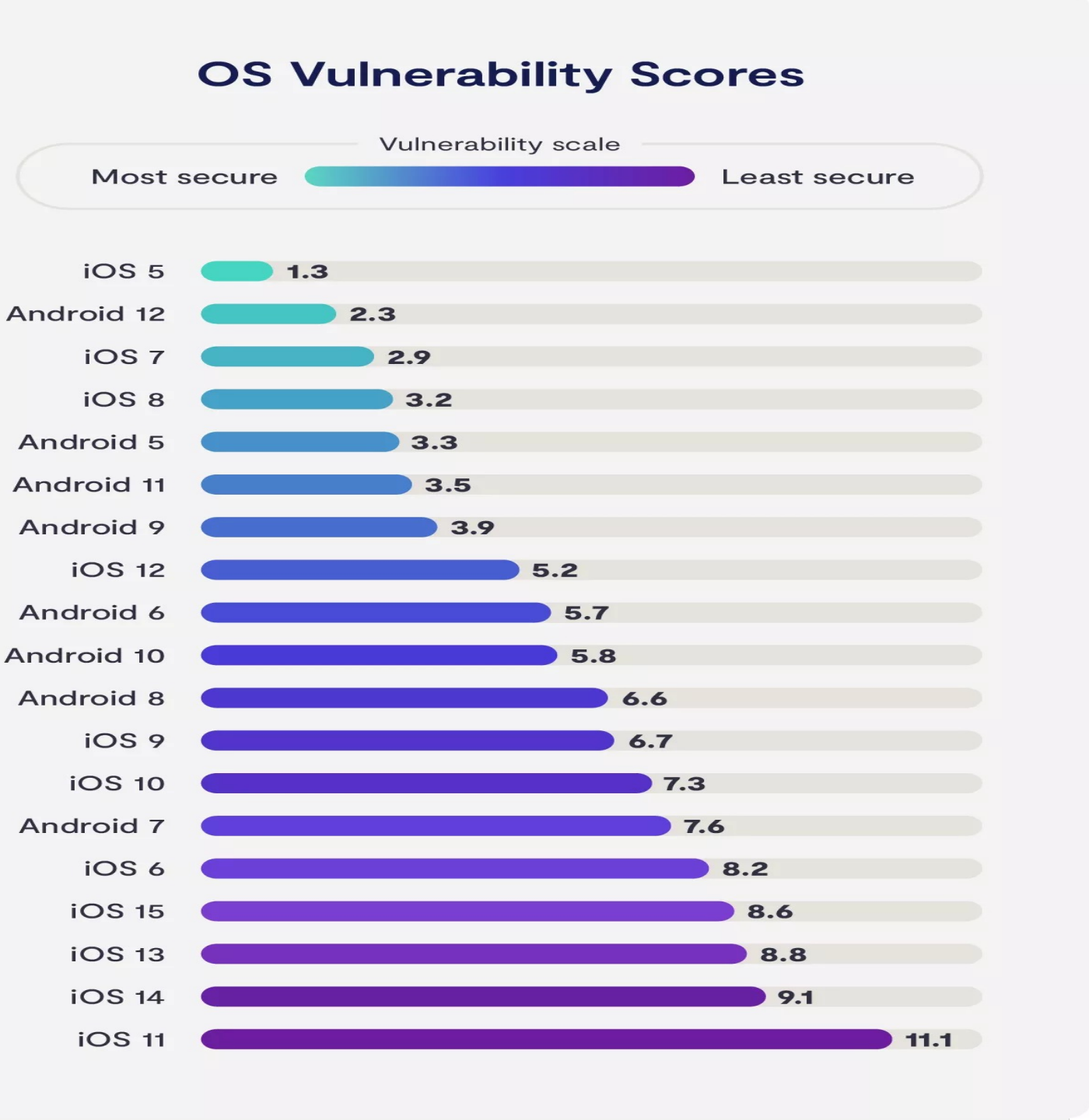
Secure booting process

- During the booting process, iOS uses a mechanism called "**secure boot chain**" to ensure that the low-level software is not compromised and iOS is running on a validated iOS device. Each step in secure boot chain verifies if the next step of chain is valid and signed by Apple. The booting process will only proceed to the next step of chain if the verification succeeds.
- When you turn on an iOS device, the processor first executes the code from Boot ROM.
- The code in Boot ROM is created during chip fabrication, hence it is trusted and immutable. The code in Boot ROM also contains the Apple Root CA public key, which will be used to verify if Low-Level Bootloader (LLB) is signed by Apple.
- If LLB is valid, the processor will run the next-stage bootloader, iBoot, which will in turn verify and run the iOS kernel.

Secure Enclave

- The Secure Enclave is a coprocessor for Apple's A-series processor. It has its own secure boot separated from the application processor, communication between it and the application processor is highly encapsulated. Its tasks include key management, processing cryptographic operations and maintaining the data integrity.
- Each Secure Enclave comes with a unique ID (UID) during the fabrication. Other parts of the system don't have access to UID, neither does Apple. UID is used to encrypt the Secure Enclave's memory space and data of files stored in the file system.
- The Secure Enclave is also responsible for decrypting and processing the fingerprints received from the Touch ID, verifying if the coming fingerprints match the registered fingerprints. The application processor forwards the fingerprints data to the Secure Enclave. Because the fingerprints data is encrypted with a session key between the Secure Enclave and the Touch ID, the application processor can't read it.

iOS 5 and 7 and Android 12 were the most secure while i**OS 11, 13, and 14 were the least secure**.



iOS vs. Android: Which is more secure in 2022?

- most secure operating system is iOS
- Ultimately, what gave iOS the win was its ability to deliver frequent updates to almost all of its recent devices. While Android has had secure updates over the years, its ability to deliver those updates quickly is limited
- Whether you own an Android or an iPhone, however, you can never be 100% safe from security breaches and zero-day attacks.
- Make sure that you're always proactive about using security solutions such as VPNs and spyware removal tools.

https://clario.co/blog/ios-vs-android-security/

What do iOS and Android security have in common?

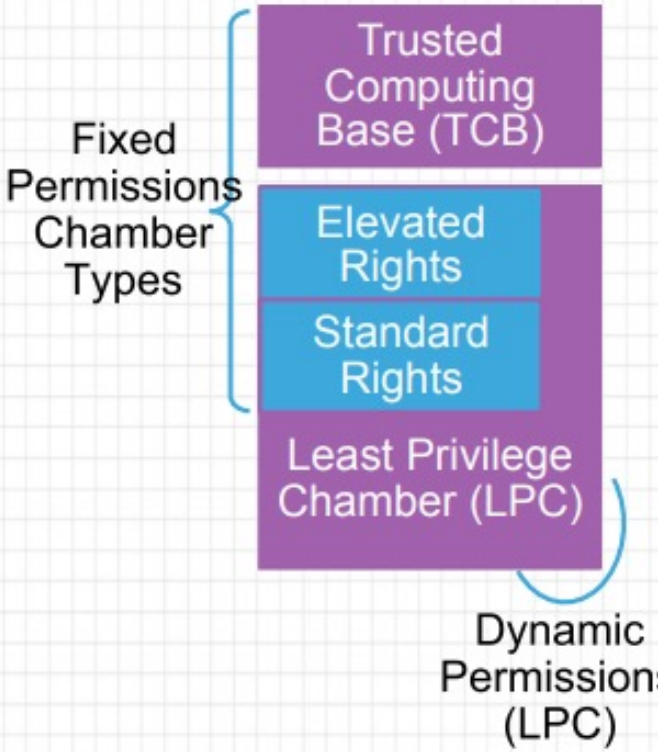
- Both iOS and Android have similar built-in features, including virtual sandboxes that limit the damage that malware apps can do. iOS drive encryption comes standard while Android users must enable this feature.
- Both OS fully support VPN encryption, which is especially important for mobile devices (NordVPN provides top-of-the-line security to both iOS and Android devices.

	iOS	Android
App marketplace security	✓	✗
Secure manufacturing process	✓	✗
Security updates	✓	✗
Support for third party security software	✓	✓
Source code security	✓	✓
Most widely used	✗	✓
Most secure OS	✓	✗

Windows Phone security

- Windows mobile OS is similar to iOS in that a human reviews and approves all apps submitted to the store, helping prevent malicious applications gaining access to the Windows Store. Unlike Android, there's no need to consider dedicated anti-virus and anti-malware software.
- currently Windows is the least utilized mobile OS of the three, which definitely plays in its favor as it is less of a target.
- Microsoft's Windows Phone platform is the safest mobile operating system available to businesses while Android remains a haven for cyber criminals.

Windows Phone 7 security model



Policy system

Central repository of rules
3-tuple {Principal, Right, Resource}

Chamber Model

Chamber boundary is security boundary
Chambers defined using policy rules
4 chamber types, 3 fixed size, one can be expanded with capabilities (LPC)

Capabilities

Expressed in application manifest
Disclosed on Marketplace
Defines app's security boundary on phone

When it comes to security on Windows mobile devices, there are two important principles at work:

- the idea of least privilege and
- the concept of isolation.

Least privilege means that rights and permissions for users or developers are restricted to only the minimum necessary to complete the task. In short, rather than having free rein of all of a Windows Phone's processes, an app developer will only have access to those areas required to reasonably perform a task.

Isolation is the notion that phone elements and processes have boundaries within which they must operate, without infringing into the boundary of any other element or process.

- To construct this type of security model, Windows Phone developers turned to the idea of **security chambers**.

Windows Phone 8 security model

Similar to WP7

WP8 chambers are built on the Windows security infrastructure

Services and Application all in chambers

WP8 has a richer capabilities list

Comparison

	iOS	Android	Windows
Unix	x	x	
Windows			x
Open market		x	
Closed market	x		x
Vendor signed	x		
Self-signed		x	x
User approval of permissions		x	7-> 8
Managed code		x	x
Native code	x		

	Android	iOS	Windows
Memory Management			
Memory usage	High	Low	High
Memory used for App handling	RAM	RAM	RAM + VM
Process running in background	Not Efficiently	Efficiently	Not Efficiently
Use of Garbage Collector	Yes	No	Yes
Background Processes	Do not freeze	Freeze	Suspend
To increase process speed	Uses internal memory	Don't use internal memory	Uses internal or virtual memory
Interface	User Friendly	User Friendly	Not User Friendly
Increase in Memory demand	Lag in app handling	No lag in app handling	Lag in app handling
Shortage of Memory	May kill some processes	Freeze background processes	Uses Virtual Memory
Capable of loading large number of apps	No	No	Yes
Security			
Arrival of new process	May kill existing process	Freeze some processes	No other processes will be affected
Utilities used	Own and third party	Own	Third Party Mostly
Issue Occurrence	Use patches	Use patches	Deliver updates
Rooting	Allowed	Not allowed	Not allowed

References:

Books:

1. Hacking Exposed Mobile: Security Secrets & Solutions 1st Edition, Kindle Edition, by Neil Bergman, Mike Stanfield, Jason Rouse, and Joel Scambray
2. Hacking Exposed Web Applications, 3rd edition, Joel Scambray, Vincent Liu, Caleb Sima, Released October 2010, Publisher(s): McGraw-Hill

Video Lectures :

1. https://www.kaspersky.com/resource-center/threats/android-vs-iphone-mobile-security
2. https://www.diffen.com/difference/Android_vs_iOS

Reference Links:

1. https://www.tutorialspoint.com/mobile_security/mobile_security_windows_phone_os.htm
2. https://logrhythm.com/solutions/security/zero-trust-security-model/
3. https://crypto.stanford.edu/cs155old/cs155-spring14/lectures/17-mobile-platforms.pdf

THANK YOU