**Advanced Topics in Operating Systems Security Vulnerabilities and Cyber Defenses Assignment 2**

**Part1:**

The Android OS uses a mechanism called the Application sandbox to enforce security. This mechanism is a Linux, kernel-level Application Sandbox. The Application sandbox provides Android with several key security features, including:

* **A user-based permissions model**: In this mechanism, The Android platform takes advantage of the Linux user-based protection as a means of identifying and isolating application resources. The Android system assigns a unique user ID (UID) to each Android application and runs it as that user in a separate process. This approach is different from other operating systems (including the traditional Linux configuration), where multiple applications run with the same user permissions.
* **Process isolation:** Applications get a dedicated part of the filesystem in which they can write private data, including databases and raw files. The Application Sandbox Prevents user A from reading user B's files. Ensures that user A does not exhaust user B's memory. Ensures that user A does not exhaust user B's CPU resources. Ensures that user A does not exhaust user B's devices (e.g. telephony, GPS, Bluetooth).
* **Extensible mechanism for secure IPC:** apart from the standard Linux IPC mechanisms Android also provides new IPC mechanisms:
  + Binder: A lightweight capability-based remote procedure call mechanism designed for high performance when performing in-process and cross-process calls.
  + Services: Services can provide interfaces directly accessible using binder.
  + Intents: An Intent is a simple message object that represents an "intention" to do something. For example, if your application wants to display a web page, it expresses its "Intent" to view the URL by creating an Intent instance and handing it off to the system. The system locates some other piece of code (in this case, the Browser) that knows how to handle that Intent, and runs it. Intents can also be used to broadcast interesting events (such as a notification) system-wide.
  + ContentProviders: A ContentProvider is a data storehouse that provides access to data on the device; the classic example is the ContentProvider that is used to access the user's list of contacts. An application can access data that other applications have exposed via a ContentProvider, and an application can also define its own ContentProviders to expose data of its own.
* **The ability to remove unnecessary and potentially insecure parts of the kernel**

this security model also extends to operating system applications. All software above the kernel, such as operating system libraries, application framework, application runtime, and all applications, run within the Application Sandbox. Thus,both ART and native applications run within the same security environment, contained within the Application Sandbox. this allows no restrictions on how an application can be written that are required to enforce security; in this respect, native code is just as secure as interpreted code. this security model also extends to operating system applications.

In contrast to what we learned in class where sandboxes like chromes sandbox are maintained within the application and need the explicitly written and supported by the app developer. Here the broker process is the kernel itself.

In Android, personal information cost sensitive APIs and Sensitive Data Input Devices are secured by using protected APIs. When trying to access a resource via a protected API the requesting users (process) privileges are checked and access is allowed only if the user has sufficient rights. Applications that choose to share information can use Android OS permission checks to protect the data from third-party applications. Upon installing any application, the user in prompted with a dialog showing all the permissions this app requires. The user in turn must allow these permissions in order for the app to be installed.

The above information is taken from:

<https://source.android.com/security/overview/kernel-security>

<https://source.android.com/security/overview/app-security>