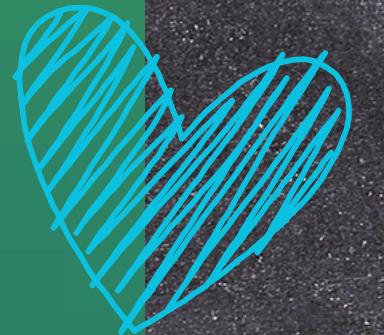
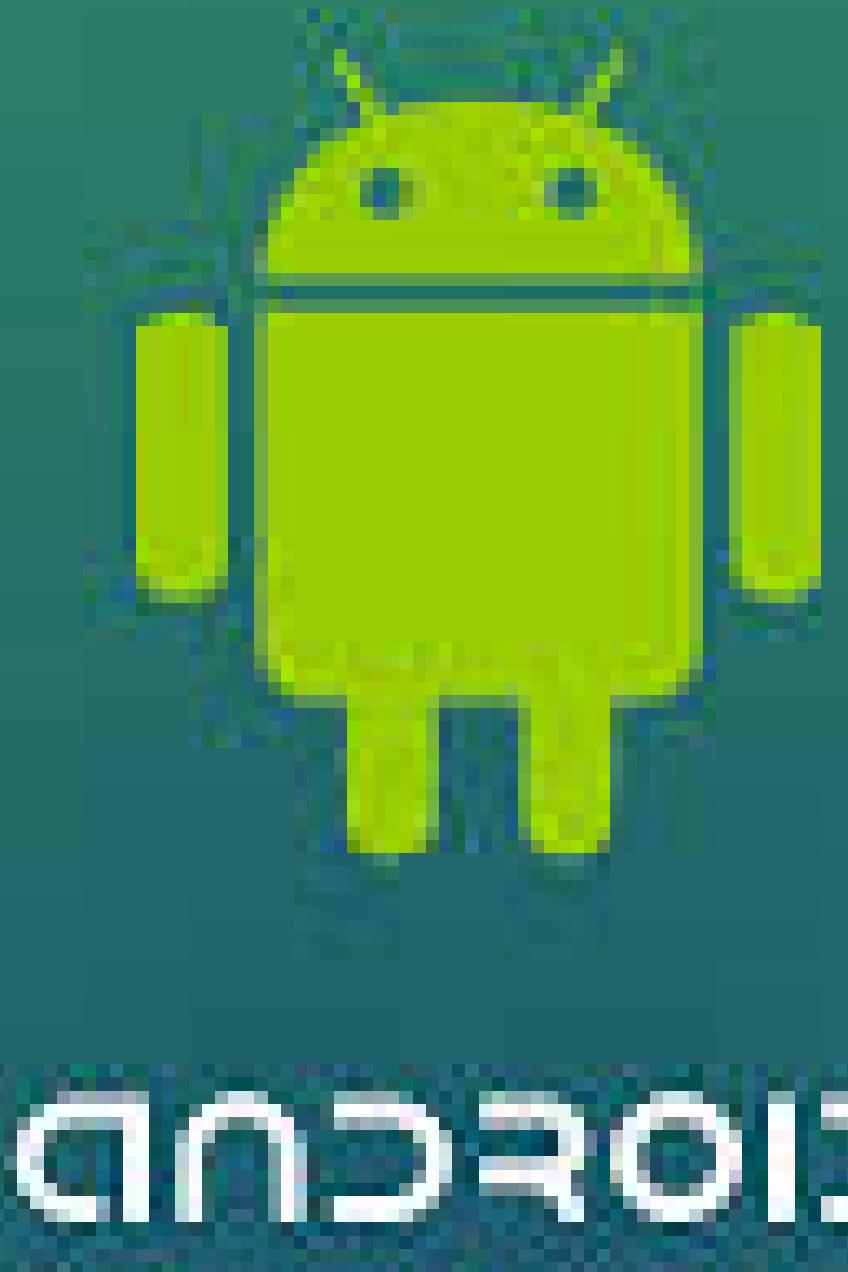


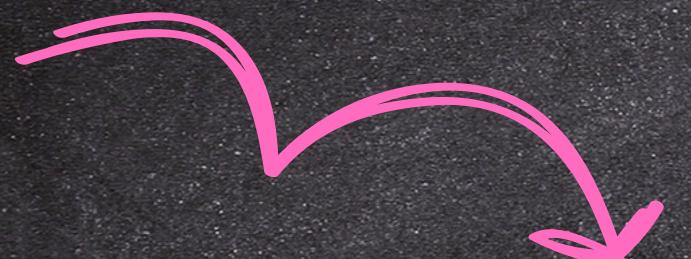
Task Hijacking in Android

ANSHIKA VERMA



- Open-source operating system
- Based on the Linux kernel

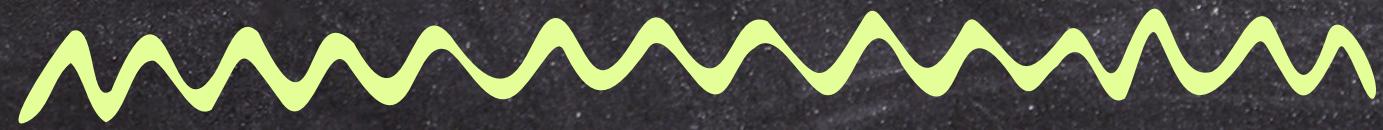




Intro

Anshika Verma

- Associate Security Analyst
@CloudSEK





Objective



Terms to know



Activity

Represents a single screen with a user interface (UI) in an Android application.

Task

A collection of activities that are grouped together and managed as a unit by the Android system



Terms to know



Intent

A message or communication object that is used to request an action or to pass data between different components in an Android application

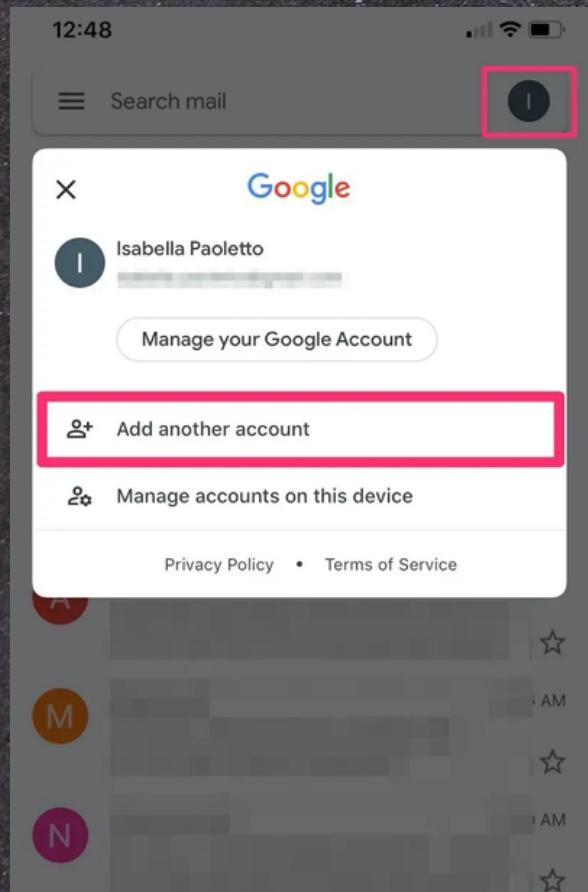
Instance

When we talk about instances in the context of launch modes, we are referring to instances of activities. Each time an activity is created and added to the task stack, it is an instance of that activity

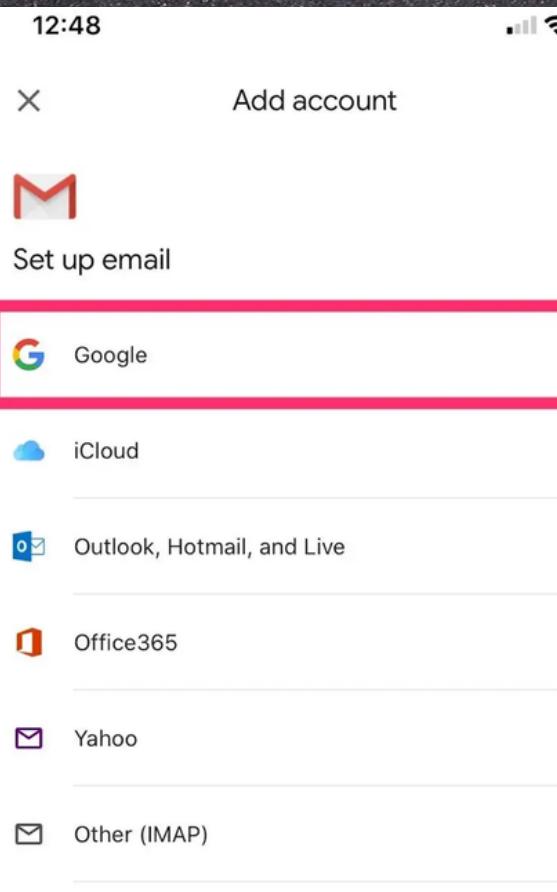
Example

Consider Gmail login task.

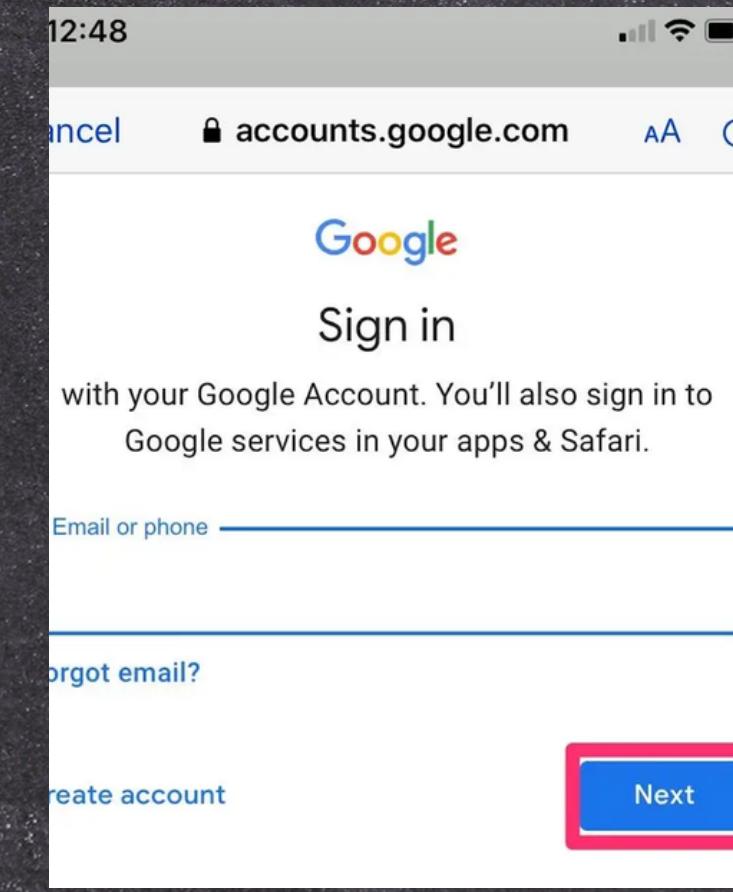
Activity 1



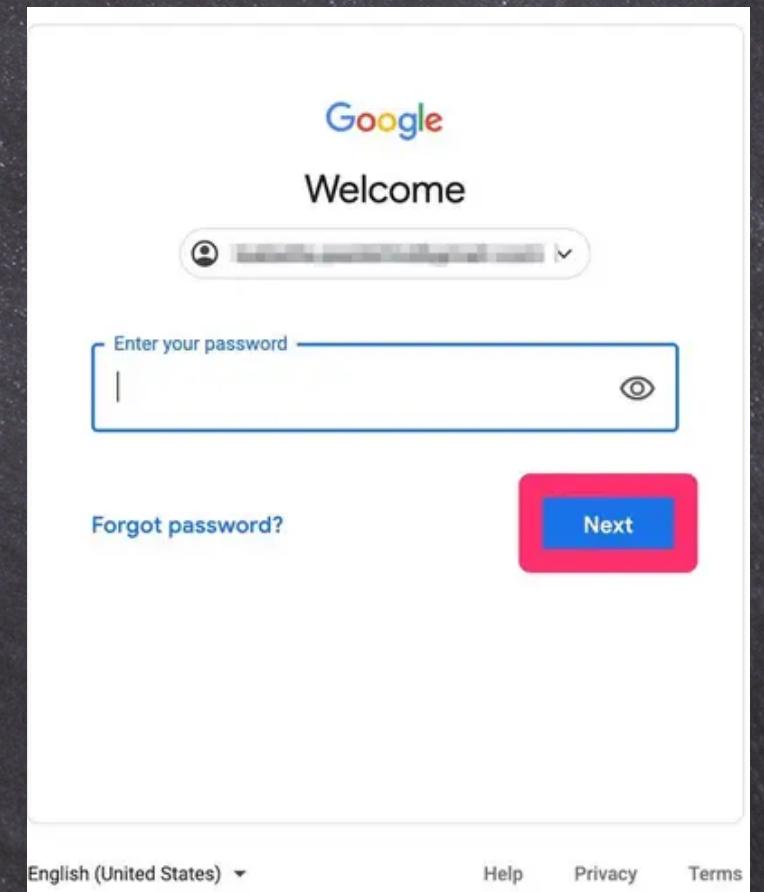
Activity 2



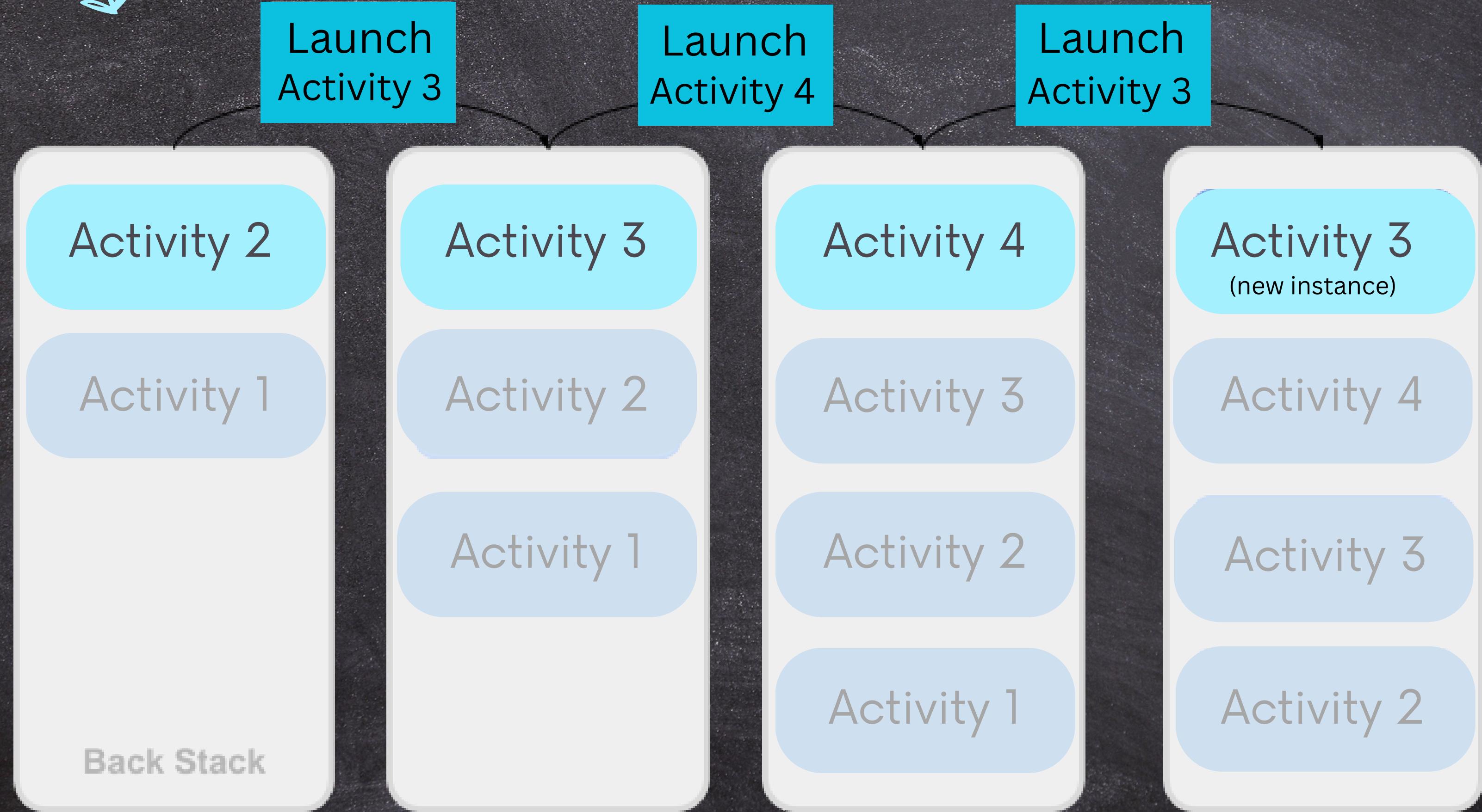
Activity 3



Activity 4



Task - A



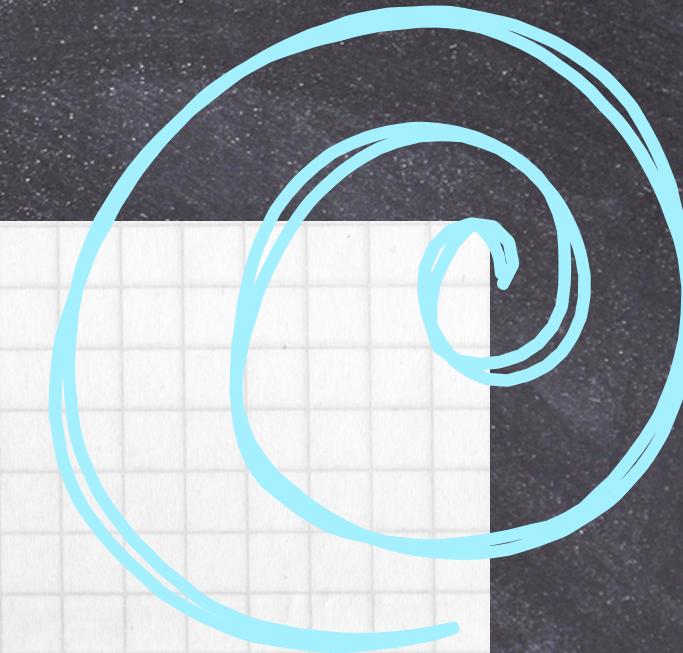
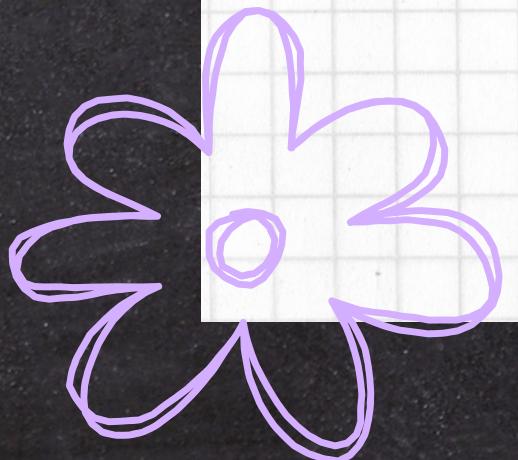
Launch modes

standard

singleTop

singleTask

singleInstance



Standard

: Home Screen -> Product Details -> Cart
: Home Screen -> Product Details -> Cart -> Product Details

Single Task

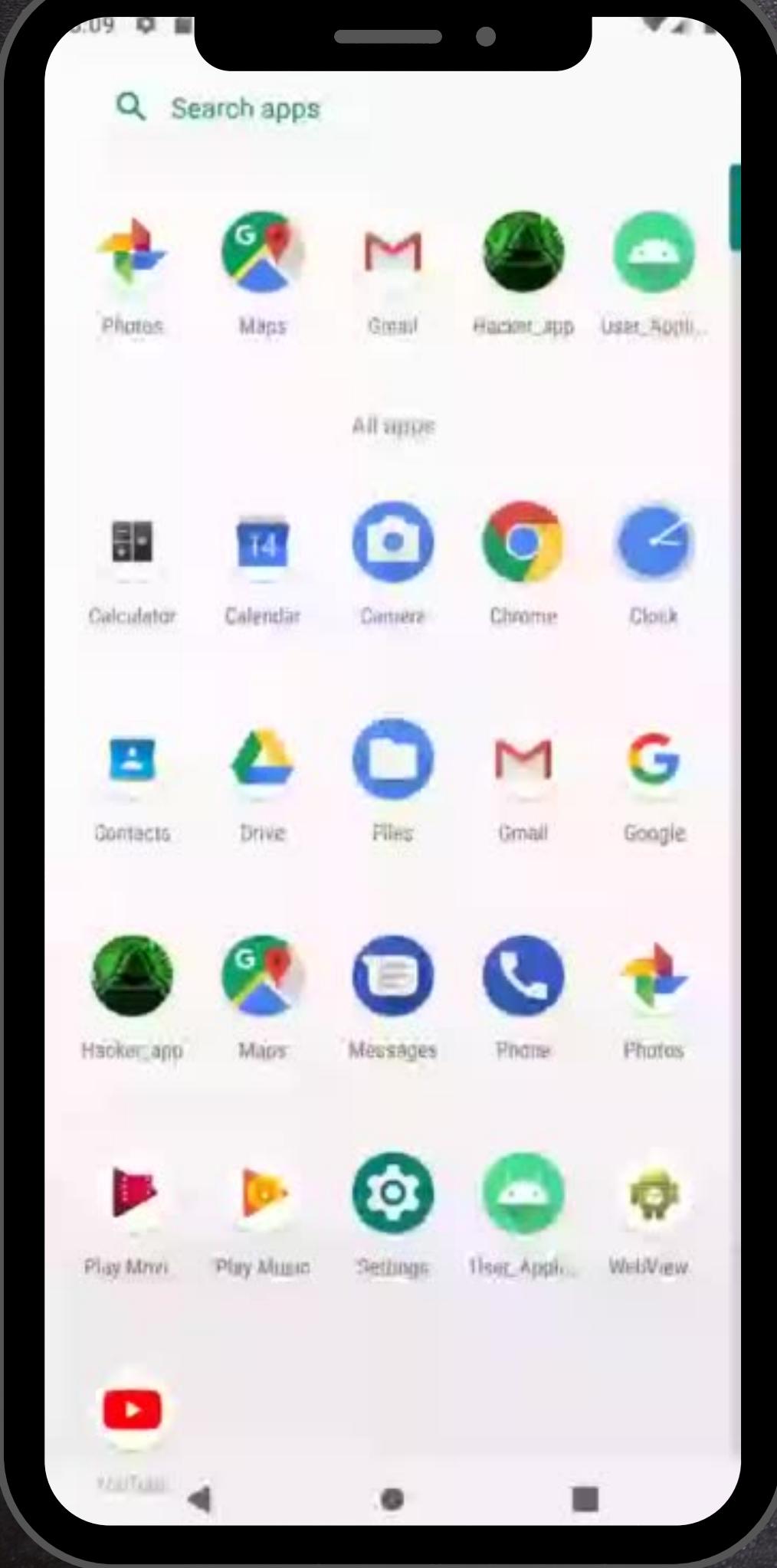
: Home Screen -> Product Details -> Cart
: Home Screen -> Product Details -> Cart -> Product Details

Single Top

: Home Screen -> Product Details -> Cart
: Home Screen -> Product Details -> Cart -> Product Details

Single Instance

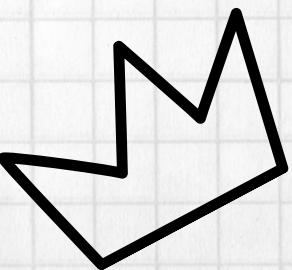
: Home Screen -> Product Details -> Cart -- (task A)
: Home Screen -> Product Details -- (task B)



Task Hijacking

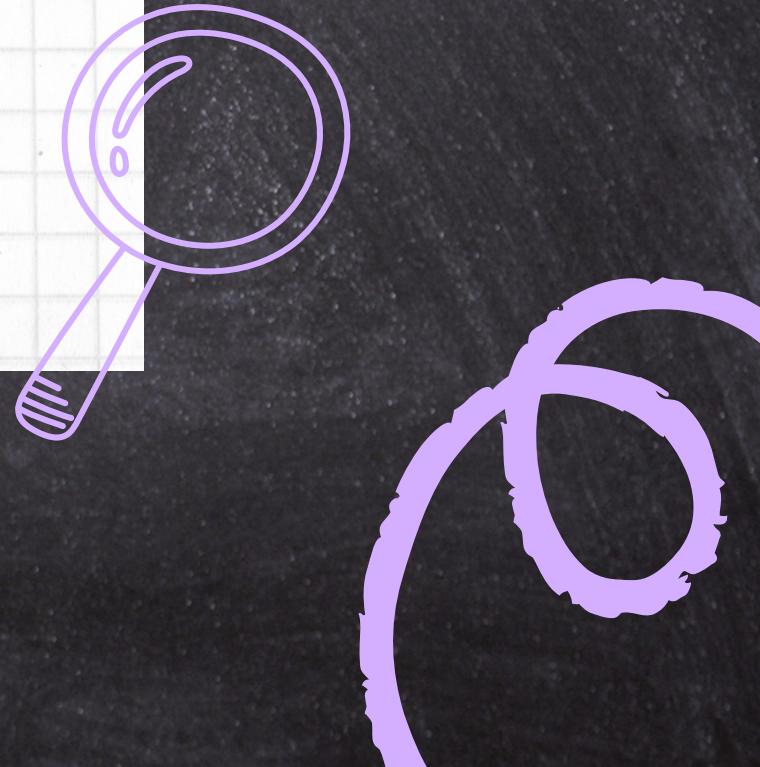
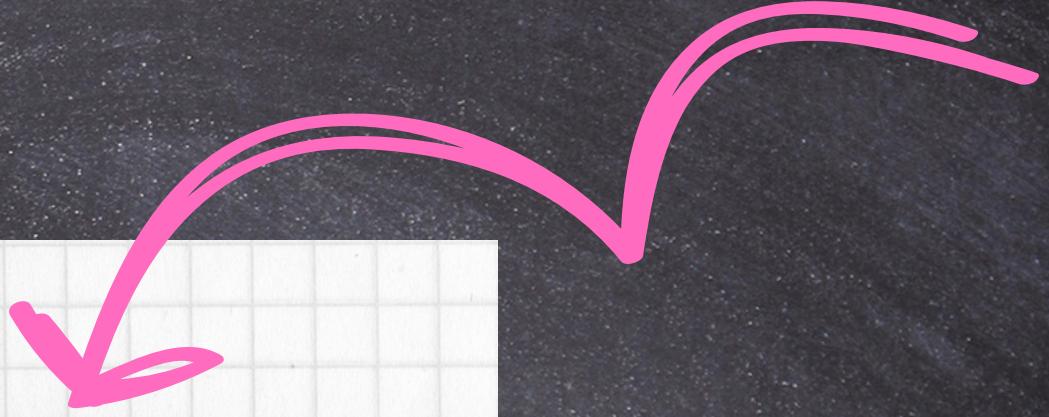
Task interception or task stealing is a security vulnerability that occurs when a malicious app or component takes control of another app's task or user interface (UI) without the user's knowledge or consent





Mitigations

- Regularly Update Devices and Apps
- Download Apps from Trusted Sources
- Be Cautious of App Permissions
- User Awareness and Vigilance
- Implement Secure Coding Practices
- Regular Security Testing



THANK YOU

SPECIAL MENTIONS

NULL
SAURABH
AARUSHI KOOLWAL
PAWAN KARTHIK M
DANIEL DOMINIC
CLOUDSTERS