Mobile Security

File Permissions - Local Storage

By default apps are given their own local directory for file storage which usually resides in "/sdcard/Android/data/ASSIGNED_APP_DIR" or "/data/user/0/ASSIGNED_APP_DIR"

This directory is given by default so that each app can have it's own directory for file storage; removing issues pertaining to apps interacting within the same storage space

This level of default file system permissions is good enough for most applications that are created.

File Permissions - External Storage

With the external storage file permissions developers have read and write access to nearly all files on a given android device. (/storage)

Giving an app storage permissions will allow the app to have access to almost all of your personal files and data that is located on your device.

External storage permissions is used by approximately 59% of applications that are deemed to be malicious in some way which make up around 24% of the market.

External Storage Vulnerabilities

- Steal personal data files from device (Confidentiality)
  - Datamine your personal files and upload analyzed information to an external server
  - Leak personal data to the public
  - Steal private information (Blackmail)
  - Potentially steal credentials for other applications / websites
- Write malicious payloads to your device (Integrity)
  - Store payloads for later use
  - Replace a harmless file used by the android OS with a malicious file
- Move/Delete files from the device (Availability)
  - Compromise availability by deleting important personal documents

External Storage - When/Why is it needed?

External storage is needed when files need to be written to a user directory such as ~/Music, ~/Downloads, ~/Photos, etc...

This is one of the most overused permissions in the google play store as almost every application requires this permission even though most of these apps can get away with their own local storage space. Approximately 59% applications deemed malicious use this permission, the most common reason for the malicious use of the following permission is to data mine a user's device.

In terms of legitimate uses, applications such as web browsers, camera applications and cloud storage applications need this permission in order to function properly.

Android NDK - C/C++ Support

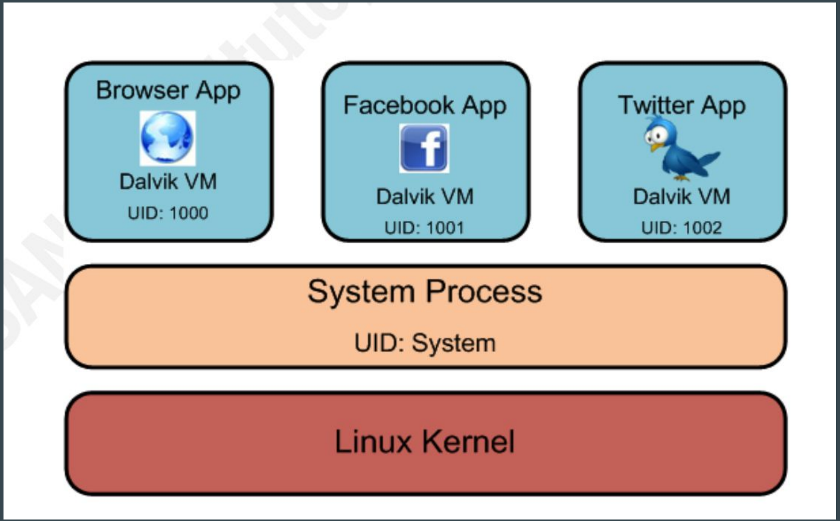
Released in 2009

Gave developer’s C and C++ support for real time applications that are time sensitive. Most of these applications included support for game libraries such as OpenGL or Vulkan which offer a massive amount of performance to push mobile gaming to the next level for those that are into that sort of thing.

It also allowed developers to utilize their previously written libraries on mobile devices.

Android NDK - System Calls

- Android applications are run using the Dalvik VM in their own memory space. System calls can be used to get access to the system layer.



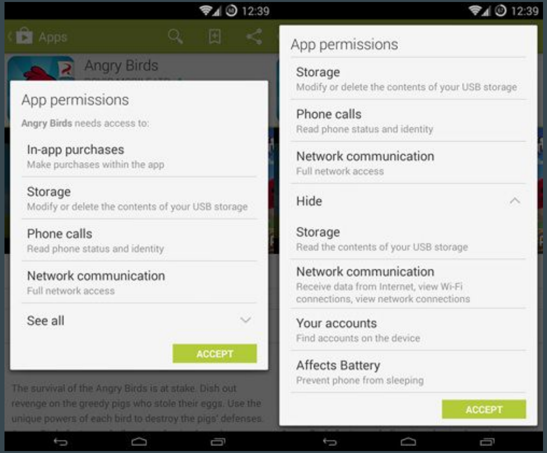
- Using the exec (java) and pipe, popen/pclose (c/c++) commands we can execute system commands that are available to us on the android device.
- To find which system utilities are available on a given device we can simply run “ls /system/bin” and “ls /system/xbin/” as well as “toybox help” to give us a list of the utilities at our disposal.
- Google has no available documentation of the commands available on an android device so the only way to figure out what some of the given commands do is through community forums as well as trial and error.
- Since the application is run in its own Dalvik VM, the system calls still have to go through the same application permission layer (ex: without storage permissions we can’t get access to /storage but we still have read access on / and /system).

System Calls - Threats

- Excessive Permissions
  - Giving the application a large number of permissions increases the risk that one of the system calls that require a given permission can be used maliciously.
- Root Access
  - If the device is rooted and the user has given the application the required permissions, the application will have access to all system commands.
  - There is also a reported bug that allows developers to get root access without the device being rooted through privilege escalation, devices past late 2016 which are up to date with the latest security patches are safe from this vulnerability.
- Background Processes (Threads)
  - An application has the ability to spawn an activity or system utility as a background process which the user would not be able to detect unless they had a way to list the currently running processes.

Excessive App Permissions

- Bad or malicious programming of apps can lead to unnecessary consequences
  - Authenticate Accounts
  - Read log data
  - Access and modify locally stored data including external storage
  - Access contacts
  - Read social streams
  - Battery life and resources
  - Camera access


12:39

Apps

Angry Birds

App permissions

Angry Birds needs access to:

In-app purchases
Make purchases within the app

Storage
Modify or delete the contents of your USB storage

Phone calls
Read phone status and identity

Network communication
Full network access

See all

ACCEPT

The survival of the Angry Birds is at stake. Dish out revenge on the greedy pigs who stole their eggs. Use the unique powers of each bird to destroy the pigs' defenses.

12:39

App permissions

Storage
Modify or delete the contents of your USB storage

Phone calls
Read phone status and identity

Network communication
Full network access

Hide

Storage
Read the contents of your USB storage

Network communication
Receive data from Internet, view Wi-Fi connections, view network connections

Your accounts
Find accounts on the device

Affects Battery
Prevent phone from sleeping

ACCEPT

Location Permissions - Last Known Location

- Android provides a function to use the location services API and retrieve the Last Known Location from a mobile device
  - Returns a location object used to retrieve latitude and longitude
  - Typically used to retrieve weather temperature
  - Geolocation Snapchat filters
  - Let Domino's know where you are at 2 am
- Can also be used maliciously
  - Sell location data to companies
  - Invasion of privacy

Abuse of Camera Service

- Approximately 86% of android malware comes in the form of repackaged apps[1]
  - Function as the original legitimate app while performing additional malicious activities
- In a survey of over 758 of the top apps[2] on Google play, 184 of them request camera permission (~25%)
- Access to the camera device is granted to any member of the camera group (any app given camera permission)

Image source:
http://weheartit.com/entry/35697047

1. Y. Zhou and X. Jiang. Dissecting android malware: Characterization and evolution. In Security and Privacy (SP), pages 95–109. IEEE, 2012.
2. See next page

Vulnerability - Transplantation attack[1]

Goal: Spy on user while bypassing API auditing, AV and mobile device management

1. Gain access to camera drivers by being part of camera group
    a. Repackage legitimate app with camera access
2. Load .so libraries into app's address space
    a. *libcameraservice.so* (system libraries) , *libhardware.so* , *libcamera.so* (hardware abstract layer)
    b. Can be done statically (need headers), or dynamically (*dlopen()*, *dlsym()* from Android NDK)
3. Use JNI calls to bridge the Java -> native code gap

Step 2 allows the malicious app to access the hardware libraries at a low enough level bypass the API calls, and device logging that triggers auditing/antivirus.

1. Zhang, Z., Liu, P., Xiang, J., Jing, J., & Lei, L. (2015, March). How Your Phone Camera Can Be Used to Stealthily Spy on You: Transplantation Attacks against Android Camera Service. In Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (pp. 99-110). ACM.

Hiding The Activity (from the user)

1. Hide application interface
    - Make it a service - no UI, runs in background
2. Hide picture window
    - Can be made 1x1 pixel without affecting saved picture in any way
3. Prevent flash/shutter sound
    - Can be suppressed via functions in the .so files
4. Transmit
    - Easy access to data/wifi and small transfers virtually unnoticeable
5. Prevent battery drain
    - Firing off about 3 dozen pictures at intervals through daylight hours used similar battery power to 10 minutes of someone playing a game

http://www.emoji.co.uk/view/12341/

Effectiveness

- ~45% success rate over wide variety of devices and OSes
  - Many failures due to difference in picture taking process flow

- Escaped both installation and run-time detection for all 7 antivirus' tested
  - New attack -> new signature; allowed bypassing installation scanning
  - AV don't view loading external .so as malicious, nor running of native code
- Device Administration preventing picture taking completely ineffective
  - Disable picture-taking APIs, this vulnerability attacks at a lower level