New Media Rights
3405 Kenyon St., Suite 402, San Diego, CA 92103
Tel: (619) 591-8870 Fax: (619) 696-7477
web: www.newmediarights.org e-mail: support@newmediarights.org

David O. Carson
General Counsel
U.S.Copyright Office
P.O. Box 70400
Washington, DC 20024

Dear David,

Please find our responses to your questions regarding follow-up to below.

1) Please provide technical details on how Google's Android OS restricts access to third party applications. (for all witnesses)

The Android OS restricts third party apps in at least two key ways.

a) Phones that limit users from installing apps not obtained from Google Play

The first way is illustrated through the fact that recent phones like the HTC Aria and Motorola Backflip have measures in place by the carrier that effectively bar third party apps all together. In this case, the carrier has removed the option of allowing apps from unknown sources from being installed the OS, effectively limiting the device to only running software downloaded through Google Play (the Android app store) a "known source." #

b) Basic technical restrictions that restrict third party applications

On a software level, the Android OS keeps applications isolated in a "sandbox" which both restricts their access to components of the device, and also protects the rest of the device from unstable programs. This inherent level of restriction is built into the Android OS for all apps, unless a program has root level access. When developers are creating an application, the Android Software Development Kit gives them access to deep levels of a phone's functioning, including access to the phone's hardware, settings and user data. An application's ability to access these functional components is dependant upon the permission given to that app during installation. Android provides pre-determined permission levels to be assigned to apps, like the "normal" access level, or the "dangerous" access level, which restricts an app's ability to send text messages or access contacts. Most important is the "system" permission level which gives access

to system files and other normally restricted components, and is only available to apps on phones which have been rooted. # Generally speaking, consumer apps that require the system permission level are either unusable with a non-jailbroken phone, or barred from Google Play.

We can see how these restrictions are applied when looking at the framework of the Android operating system. The functioning of the OS is built layer upon security layer, with each layer functioning independently from one another, protecting various elements of the OS. This creates a unique and isolated space for applications to run, where if they are insecure or unstable, they won't affect the functioning of other elements of the phone.

The foundational layer is the system and kernel level security. This level effectively houses, protects and provides a runtime environment for basic drivers like display, bluetooth and keypad drivers. It is a foundational layer because it houses the needed instructions for the phone's hardware components, typically which other applications build off of. A rooted device makes available access to this level of permissions in certain circumstances.

The next level contains the general and Android specific runtime libraries. These important files contain the code for many of the functions specific to the Android OS and the phone itself. Built upon this layer is the application framework, and the applications themselves. As you go higher up the layers, more permission is needed to access lower levels, effectively isolating applications and limiting their access to private user information and core phone drivers and files.

The isolation of applications creates what is commonly referred to as an "application sandbox." In its simplest terms, the application sandbox is a place where even buggy, unstable, or even dangerous applications can be run without significant risk to damaging core components of the phone. However, because this is an artificial space for applications to run, the Android OS has almost total control over what applications are allowed to do and not do within the sandbox.

From within the application sandbox, the Android OS innately protects the following functions (APIs): camera function, location (GPS) function, bluetooth function, telephony and text message functions, and network data connections. Most all of these functions can be accessed by giving the application appropriate permissions. #

However, there are some functions that are enhanced by having root access to the phone. For example, many users wish to backup the data on their phone. Yet because of Android's internal restrictions, non-rooted users are limited by the OS in terms of what data they can and cannot access. While iOS has a complete system backup offered through iCloud, Android does not have such a comprehensive service or option. Many of the files a user would want to backup are either hidden or protected by Android. Fortunately for the rooted user, there is "Titanium Backup," which provides a complete backup of all data. This allows users who need to restore their phone to essentially pick

up where they left off with a full backup rather than risk the complications of a partial backup.

Similarly, many users want to be able to block incoming calls to their phone. While there are applications available through Google Play that allow call blocking, because a non-rooted Android phone does not grant these apps "system level" permission, the phone must allow the call to come through before the app can block the call. Without root permission, the phone must first attempt to ring, and then have the app step in and block the call. However, a rooted Android phone running an app like "Root Call Blocker Pro" can intercept the call before the phone sends any commands, and block the incoming call before the phone begins to ring.

In summary, some carriers are now restricting installation of applications found outside of the Google Play app market, similar to the situation with Apple and the App store where users are limited to a single choice in app markets. In addition, the Android OS quarantines applications into a sandbox where their ability to access "system" level functionality is limited for various reasons. This limitation can limit legitimate applications and functionality, and the only way for users to obtain access to those legitimate applications and that functionality is to allow jailbreaking of the device.

3) At the June 5 hearing, the Business Software Alliance alleged that jailbreaking mobile devices leads to/results in piracy of copyrighted applications. Please discuss the relationship between jailbreaking and piracy, and whether this is relevant to this class of works. In this context please discuss the accuracy of reliability of the articles and links previously submitted to the Office discussing apps and piracy (for proponents and opponents)

The June 7th email from Jesse Feder of BSA to Ben Golant provided articles and links, but these articles and links are not relevant to the present discussion on piracy and jailbreaking. The links simply provided references for statistics about number of users of various devices and number of apps, and have little to do with piracy.

However, generally, speaking the link between piracy and jailbreaking is thin.

A jailbroken phone is a phone with less restrictions. Any time one lessens the restrictions on something, it makes improper and unintended uses easier for someone to choose to make. However, the choice to pirate software is a choice made on the user's end, whether the phone is jailbroken or not. The emphasis here is on the user's choice. There is nothing inherent within a jailbroken phone that makes piracy more likely. Someone who installs a jailbreaking application on their phone to remove bloatware or unwanted applications, may have no idea where to find pirated Android software on the internet. One is a legitimate use, while the other is an abuse.

Moreover, many of the sites which promote user's jailbreaking their phone, strictly

forbid content that is illegal. These sites grow communities of developers, who with the knowledge partially gained from jailbreaking a phone, produce “homebrew” original works specifically for jailbroken Android phones. The jailbreaking apps and other programs created by these communities of Android enthusiasts have sometimes hundreds of hours of work invested into creating them. The risk of being shutdown over a legal claim keeps most of these communities from allowing illegal material.

This cloisters software piracy on jailbroken smartphones to a small niche only occupied by fringe users in a grey market. By making access to copyrighted material more difficult, users are forced to actively seek out infringing material, which itself is often hidden beneath layers of misleading web links and requires an adept user to navigate correctly. The distinction to be made is that, though a phone is jailbroken, it does not suddenly open up a secret world of copyrighted material ripe for piracy. Accessing and pirating material on a jailbroken phone is still a complex task requiring commitment and a particular level of technical expertise. It is NOT a one-click experience, like for example purchasing a song from iTunes.

This being said, a user committed to piracy is going to pirate copyrighted material whether it’s on a jailbroken phone, a phone that simply doesn’t require jailbreaking, or a personal computer. Legislating against jailbreaking to combat piracy, is a little bit like legislating against photocopy machines because they could be used to make illegal copies of a book.

4) EFF has recently proposed a definition of “Tablet” for this class of works that reads as follows:

> a) a personal mobile computing device, typically featuring a touchscreen interface,
>
> b) that contains hardware technically capable of running a wide variety of programs,
>
> c) that is designed with technological measures that restrict the installation of modification of programs on the device, and
>
> d) is not marketed primarily as a wireless telephone handset.

Assuming for the purpose of this inquiry that tablets are part of the proposed class, please comment on the appropriateness of this definition. (For Mr. Neill and for Mr. Menon and Mr. Lassey; other witnesses have already commented on EFF definition.)

New Media Rights supports the proposed definition of “Tablet” as proposed by EFF.

Respectfully submitted,

Art Neill and Alex Johnson
New Media Rights
3405 Kenyon St., Suite 402
San Diego, CA 92110