James Evans Turner Technical Support Representative, Software Developer, Technology Enthusiast

Addressing Classes 3, 4, and 5

Regarding Class 3: (video game consoles)

Running unauthorized software on a game console through "jailbreaking" is understandably discouraged by game console manufacturers because it can enable software piracy. It can also enable cheating, which impacts the quality of online multi-player games (causing fair game players to experience a disadvantage). For this reason, I believe that game developers should be free to detect unauthorized software that is known to have these effects. They should be able to ban those modified systems and associated user accounts from accessing online services and resources. However, the legality of modifying hardware and software to enhance its capability through "unauthorized" software or hardware modification should be specifically protected and declared legal. Many users purchase a piece of hardware with specific functionality in mind. Sometimes, that functionality is not provided without modification. Sometimes the manufacturer's features are poorly-implemented or neglected. Worse, functionality can be unexpectedly removed with a manufacturer's update (A couple of examples: 1. Sony removed PS3 Linux support. 2. Nintendo removed MP3 support in Wii Photo Channel). Manufacturers have also been known to deliberately implement limitations to encourage planned obsolescence. Without the ability to run unauthorized software, a physically flawless piece of equipment can be rendered worthless as the manufacturer shuts down services only a few years after purchase. We should consider the growing problem of electronic waste and allow users to continue using their devices for alternate purposes, which can be enabled through jailbreaking. At the core, game systems are nothing more than computers with well-defined characteristics. Even the U.S. Dept. of Defense purchased over 2,200 Sony PS3 systems to run custom software that leveraged the system's Cell CPU to perform complex mathematical computations. Through jailbreaking, any owner can use custom software to find new purposes for the hardware.

To the point of software piracy being enabled by jailbreaking: Millions of game consoles are rendered useless after hardware component failures, especially mechanical components like optical drives and hard drives. These systems are usually thrown away and end up in a landfill, contributing to the world-wide electronic waste problem. Through jailbreaking, even devices with failed components can continue to be useful. For example: By installing a hard drive and modified memory card into a Sony PS2, it can run software even when the built-in optical disc drive stops working (a problem that has affected millions of early units). In compliance with copyright law, a backup copy of the original software can be legally created and stored on the hard disk drive. This can be done through a network connection, or by connecting the hard drive directly to a PC or a system with a working optical disc drive. Even though optical disc drive components frequently fail, some newer systems (including Sony's PS3) have protections to prevent users from using a working drive from another system.

Class 3 Summary: Intelligent, creative citizens should be expressly permitted to legally "tinker" with their own property, even when it's done without the manufacturer's blessing.

Regarding Class 4: (personal computing devices) Smartphones, tablets, and video game systems are all personal computer devices that execute software, so the points I detailed for classes 3 and 5 also apply here.

Class 4 Summary: Despite the wishes of a hardware manufacturer, we should protect the right of an owner to modify software and hardware to enable new capabilities in a device or extend the usable lifetime of that device.

Regarding Class 5: (mobile phones and tablet computers)

It is particularly important that owners should be legally permitted to modify software and hardware for mobile devices, even if it voids the warranty for these devices. There are numerous examples of accessibility problems that cannot be overcome without running unauthorized software. Here are several examples from my personal experience with Apple's iOS5, the latest software for iPhone, iPod Touch and iPad: - Though iOS5 has an excellent speech synthesis API, the iPhone does not have an option to announce the name of an incoming caller. This would be potentially life-saving for users with a Bluetooth headset or hands-free system. Often, a user must check to see if a call is important before answering. Checking this can be distracting and even deadly. Many damaged phones were dropped while removing from a pocket to identify an incoming call. - For any user of a Bluetooth device that supports A2DP, most alert sounds are inaudible because the sound finishes playing before the Bluetooth device can initialize. This is true of nearly all Bluetooth A2DP devices, and there are many ways that the software could compensate for it. However, Apple ignores the issue. - There is no way to enable the LED flashlight on the iPhone without finding and launching an app. In an emergency situation, this can waste precious time or lead to an accident. Also, a flashlight app leaves the front display active at the same time, which impairs visibility and drains the battery more quickly. - The screen brightness setting is not quickly accessible. (A) Stepping-out into sunlight can cause distraction while trying to view something. This impairment can lead to an accident or death while trying to accomplish an important task. - (B) Even if a user does not answer the phone while driving, an incoming call while driving at night can cause a blinding, bright screen that impairs the driver from seeing outside the car (the lit-up vehicle interior reflects on the windshield and diverts the driver's eyes from focusing outside). - A stolen phone cannot be tracked or recovered if the thief turns off the device.

Any of these problems can be easily addressed with unauthorized software tweaks that can only be done by jailbreaking the device. "Activator" with an extension allows me to quickly increase or decrease brightness by swiping right or left at the top of my phone screen (though I can assign many other buttons, actions, and gestures for this). Activator+SpringFlash allows me to quickly enable the light on my iPhone 4 when I need to see something in the dark. With a jailbreak tweaks, I could make it so my device only pretends to shut down, so the GPS tracking feature can be used to capture a thief and recover the device. With a jailbroken device, I can add an audio prefix for notification sounds, so A2DP devices have time to initialize and the sound can be heard. A jailbreak tweak can announce the name of an incoming caller over Bluetooth, so I don't have to be distracted or risk damage to my device to check who is calling. Through a jailbreak tweak, a Sony Bluetooth wristwatch with caller ID and music controls can work with the iPhone.

Continuing with the example of Apple's iOS devices (iPhone, iPod Touch, iPad) ...

Sometimes, a software vulnerability is discovered that allows a user to initially jailbreak a device. Such a vulnerability in the original software could also be exploited by a virus or malicious hacker. It's only a matter of time before a devastating virus takes full advantage of WiFi+Bluetooth+data capabilities to rapidly infect tens of millions of smartphones. When jailbreak developers discover a software vulnerability that can be used to jailbreak a device, they also release a software patch to close the vulnerability after the phone is jailbroken...even before Apple releases an official update to close the vulnerability. Sometimes, a user MUST jailbreak a device to secure a known vulnerability! The security problem is worsened when you consider planned obsolescence. For example: Apple never released iOS4 for the original iPhone. Even though the original iPhone had the same CPU and memory as the iPhone 3G (and the same software performance characteristics), it was quickly abandoned. Apps were updated to require iOS4, and older devices could not specifically download an older iOS3-compatible version of the app. Because the iPhone 3G received the iOS4 update, it seemed that Apple would support it for the rest of the current generation. However, the iPhone 3G was also abandoned by Apple and stopped receiving iOS4 updates even before iOS5 was released. Even worse, KNOWN software vulnerabilities are no longer being patched for iPhone and iPhone 3G. Even with the lack of support from Apple, these old smartphone devices with multi-touch, Bluetooth, and Wi-Fi are still more-desirable than an older feature-phone. Tens of millions are still in-use. Soon, Apple will stop providing software updates for the iPhone 3GS, which is still a very capable smartphone that most current users are very content with. Unofficially, an iPhone can run an open-source operating system (like Android) and continue to receive security updates and useful features. If hundreds of millions of devices are still in-use with known software vulnerabilities that will not be fixed by the manufacturer, it's important that users should be allowed to tweak these devices to protect themselves. Once again, the issue of electronic waste also cannot be ignored. It's disturbing to think that hundreds of millions of functional devices and parts would be thrown away because the manufacturer stopped supporting it.

For a non-Apple example of why it's necessary that users reserve the right to run unauthorized software: HP recently discontinued the TouchPad product line within a month of release. Most users at the time had just paid $600 or $700 for a tablet device with excellent hardware characteristics, but an uncertain future. It seemed that HP would stop supporting it. Remaining units were immediately sold-off for $99 and $150. The included WebOS software had great potential, but there were many features that did not work correctly and user interface glitches made many normal web activities impossible. Other tablets in the same price range running iOS and Android software received regular updates and had a large selection of useful software apps. There is still great uncertainty about HP's continued support for the WebOS software platform. The "CyanogenMod" team released TouchPad-specific builds of the open-source Android operating system. This allows TouchPad owners to use the latest Android software (4.0, "Ice Cream Sandwich") on their TouchPad devices, making it as useful as any other tablet in the same price range.

Class 5 Summary: It's not just a matter of convenience or accessibility. It's a matter of safety and security. We must protect a user's right to legally modify software and hardware on mobile computing devices, even if it voids the manufacturer's warranty.