Commenter: Matthew G. Muscari (muscarim@cs.rutgers.edu)

Comments on
"Exemption to Prohibition on Circumvention of Copyright Protection Systems for Access Control Technologies"

Classes of works covered: 5, 4

Summary of Arguments:

Class 5 Devices: Mobile Devices (Smartphones and Tablets)

As a graduate student of Computer Science, I have had occasion to work with the now popular Android operating system and application stack on a variety of hardware devices, including Smartphones and Tablets. Many OEM-modified Android packages explicitly disallow installation of 3rd-party software. "Rooted" or "Jailbroken" Android devices have had the weak software security lockouts implemented by device manufacturers disabled to enable installation of different versions of the Android operating system. Android, by design and explicitly by license, is built using an open-source methodology. Many of the software packages used in the Android operating system, including its modified Linux kernel, are licensed under the GNU General Public License v2 or the Apache License v2.0. Allowing device manufacturers to lock device software under penalty of law for deactivation of such locks (as per some interpretations of the DMCA) is a blatant violation of the at least the spirit if not the letter of the GNU GPL as well as the Apache License. The GPL grants the following rights to the licensee:

- Freedom 0: The freedom to run the program for any purpose.
- Freedom 1: The freedom to study how the program works, and change it to make it do what you wish.
- Freedom 2: The freedom to redistribute copies so you can help your neighbor.
- Freedom 3: The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

I strongly believe that allowing the DMCA exemption for unlocking personally owned mobile devices to lapse would be extremely detrimental to the ecosystem of application developers for the Android platform, which is poised to take over as the dominant platform for mobile devices in the United States. **The competition which has driven the rapid technological evolution and adoption of these devices will be absolutely stymied if owners are prohibited from creating development communities around them, and addition of new product features as well as fixes to existing bugs to the Android Open Source Project, which is the fundamental source of all Android packages, will be harmed irrevocably.**

Class 4 Devices: Personal Computing Devices (Including Desktop and Laptop PCs)

Microsoft stands poised to enforce locking of low-level device firmware in the next generation of desktop, laptop, and tablet computers. To obtain Windows 8 Hardware Certification, Microsoft is requiring OEMs to support UEFI secure boot on their devices, which if implemented will disallow the installation and booting of operating systems that have not been signed by the OEM-trusted Certificate Authority. Implementation of UEFI Secure Boot is vague, but in its simplest non-configurable form would display the following characteristics to the end user:

- Inability to be disabled
- Inability to add new trusted keys to the system keystore

These two attributes would completely preclude the installation of non-commodity operating systems on devices. In the worst case, if Microsoft were the only authority trusted in the keystore, these devices would only be able to boot Windows 8. If other vendors such as Red Hat were included, then officially distributed Linux or UNIX kernels would be bootable. Unfortunately, this is insufficient functionality to continue to foster the development of Linux and UNIX. The vast majority of Linux distributions are community maintained, and depend entirely on end users to develop new changes to operating system packages. **Many Linux distributions release new stable kernels several times per year, and go through dozens or hundreds of unstable kernel builds during the development process. Each build of a kernel by any individual user would require a new trusted certificate to be added to a UEFI Secure Boot enabled device's keystore, which is infeasible and unsustainable. UEFI Secure Boot is fundamentally incompatible with community-driven operating system development, and will lead to cartel-like collaboration between the major operating system vendors and hardware manufacturers if users are not allowed to circumvent such protection mechanisms.**