**OWASP Mobile Security Project:**

Dangers of Jailbreaking & Rooting

What is "jailbreaking" and "rooting"?

**iOS:** Jailbreaking is the process of modifying iOS system kernels to allow file system read and write access. Most jailbreaking tools (and exploits) remove the limitations and security features built by the manufacturer Apple (the "jail") through the use of custom kernels, which make unauthorized modifications to the operating system. Almost all jailbreaking tools allow users to run code not approved and signed by Apple. This allows users to install additional applications, extensions and patches without the control of Apple’s App Store*.*

**Android:** Rooting is the process of gaining administrative or privileged access for the Android OS. As the Android OS is based on the Linux Kernel, rooting a device is analogous to gaining access to administrative, root user-equivalent, permissions on Linux. Unlike iOS, rooting is (usually) not required to run applications outside from the Android Market. Some carriers control this through operating system settings or device firmware. Rooting also enables the user to completely remove and replace the device's operating system.

Why do they occur?

**iOS:** Many users are lured into jailbreaking to take advantage of apps made available through third party app sources, such as [Cydia](http://cydia.saurik.com/), which are otherwise banned or not approved by Apple. There is an inherent risk in installing such applications as they are not quality controlled nor have they gone through the Apple approval and application approval process. Hence, they may contain vulnerable or malicious code that could allow the device to be compromised. Alternately, jailbreaking can allow users to enhance some built in functions on their device. For example, a jailbroken phone can be used with a different carrier than the one it was configured with, FaceTime can be used over a 3G connection, or the phone can be unlocked to be used internationally. More technically savvy users also perform jailbreaking to enable user interface customizations, preferences and features not available through the normal software interface. Typically, these functionalities are achieved by patching specific binaries in the operating system.

A debated purpose for jailbreaking in the iOS community is for installing pirated iOS applications. Jailbreaking proponents discourage this use, such as Cydia warning users of pirated software when they add a pirated software repository. However, repositories such as [Hackulous](http://hackulo.us/) promote pirated applications and the tools to pirate and distribute applications.  
 **Android:** Rooting Android devices allows users to gain access to additional hardware rights, backup utilities and direct hardware access. Additionally, rooting allows users to remove the pre-installed "bloatware", additional features that many carriers or manufacturers put onto devices, which can use considerable amounts of disk space and memory. Most users root their device to leverage a custom read only memory (ROM) developed by the Android Community, which brings distinctive capabilities that are not available through the official ROMs installed by the carriers. Custom ROMs also provide users an option to 'upgrade' the operating system and optimize the phone experience by giving users access to features, such as tethering, that are normally blocked or limited by carriers.

What are the common tools used?

**iOS:** Jailbreaking software can be categorized into two main groups:

1. Tethered: Requires the device to be connected to a system in order to bypass the iBoot signature check for iOS devices. The iOS device needs to be connected or tethered to a computer system every time it has to reboot in order to access the jailbreak application, such as redsn0w, and boot correctly.
2. Un-tethered:Requiresconnection for the initial jailbreak process and then all the software, such as sn0wbreeze, is on the device for future un-tethered reboots, without losing the jailbreak or the functionality of the phone.

Some common, but not all of the iOS jailbreaking tools are listed below:

* [Absinthe](http://theiphonewiki.com/wiki/index.php?title=Absinthe)
* [blackra1n](http://theiphonewiki.com/wiki/index.php?title=Blackra1n)
* [Corona](http://theiphonewiki.com/wiki/index.php?title=Corona)
* [greenpois0n](http://theiphonewiki.com/wiki/index.php?title=Greenpois0n_(jailbreak))
* [JailbreakMe](http://theiphonewiki.com/wiki/index.php?title=JailbreakMe)
* [limera1n](http://theiphonewiki.com/wiki/index.php?title=Limera1n)
* [PwnageTool](http://theiphonewiki.com/wiki/index.php?title=PwnageTool)
* [redsn0w](http://theiphonewiki.com/wiki/index.php?title=Redsn0w)
* [sn0wbreeze](http://theiphonewiki.com/wiki/index.php?title=Sn0wbreeze)
* [Spirit](http://theiphonewiki.com/wiki/index.php?title=Spirit)

A more comprehensive list of jailbreaking tools for iOS, exploits and kernel patches can be found on the [iPhoneWiki website](http://theiphonewiki.com/wiki/index.php).

**Android:** There are various rooting software available for Android. Tools and processes vary depending on the user’s device. The process is usually to:

1. Unlock the boot loader.
2. Install a rooting application and / or flash a custom ROM through the recovery mode.

Not all of the above tasks are necessary and different toolkits are available for device specific rooting process. Custom ROMs are based on the hardware being used; examples of some are as follows:

[CyanogenMod ROMs](http://en.wikipedia.org/wiki/CyanogenMod) are one of the most popular aftermarket replacement firmware in the Android world. More comprehensive device specific firmwares, flashing guides, rooting tools and patch details can be referenced from the [homepage](http://www.cyanogenmod.com/).

[ClockWorkMod](http://www.clockworkmod.com/) is a custom recovery option for Android phones and tablets that allows you to perform several advanced recovery, restoration, installation and maintenance operations etc. Please refer to [xda-developers](http://forum.xda-developers.com/wiki/ClockworkMod_Recovery) for more details.

Why can it be dangerous?

The tools above can be broadly categorized in the following categories:

* Userland Exploits: Jailbroken access is only obtained within the user layer. For instance, a user may have root access, but is not able to change the boot process. These exploits can be patched with a firmware update.
* iBoot Exploit: Jailbroken access to user level and boot process. iBoot exploits can be patched with a firmware update.
* Bootrom Exploits: Jailbroken access to user level and boot process. Bootrom exploits cannot be patched with a firmware update. Hardware update of bootrom required to patch in such cases.

Some high level risks for rooting or jailbreaking devices are as follows:

**Technical Risks:**

1. General Mobile
   1. Some jailbreaking methods leave SSH enabled with a well known default password (i.e. alpine) that attackers can use for Command & Control.
   2. Entire file system of a rooted or jailbroken device is vulnerable to a malicious user inserting or extracting files. This vulnerability is exploited by many malware programs, including Droid Kung Fu, Droid Dream and Ikee.
   3. Credentials to sensitive applications, such as banking or corporate applications, can be stolen using key logging, sniffing or other malicious software and then transmitted via the internet connection.
2. iOS
   1. Applications on a jailbroken device run as root outside of the iOS sandbox. This can allow applications to access sensitive data contained in other apps or install malicious software negating sandboxing functionality.
   2. Jailbroken devices can allow a user to install and run self-signed applications. Since the apps do not go through the App Store, they are not reviewed by Apple. These apps may contain vulnerable or malicious code that can be used to exploit a device.
3. Android
   1. Android users that change the permissions on their device to grant root access to applications increase security exposure to malicious applications and potential application flaws.
   2. 3rd party Android application markets have been identified as hosting malicious applications with remote administrative (RAT) capabilities.

**Non-technical risks:**

1. Under the current Digital Millennium Copyright Act (DMCA), jailbreaking or rooting is termed as 'legal' in the US, which can provide some users with a false sense safety and jailbreaking or rooting as being harmless. Please refer to '[Rulemaking on Anticircumvention](http://www.copyright.gov/1201/2010/)' for more details.
2. Software updates cannot be immediately applied because doing so would remove the jailbreak. This leaves the device vulnerable to known, unpatched software vulnerabilities.
3. Users can be tricked into downloading malicious software. For example, malware commonly uses the following tactics to trick users into downloading software.
   1. Apps will often advertise that they provide additional functionality or remove ads from popular apps but also contain malicious code.
   2. Some apps will not have any malicious code as part of the initial version of the app but subsequent "Updates" will insert malicious code.
4. Manufacturers have determined that jailbreaking or rooting is a breach of the terms of use for the device and therefore voids the warranty. This can be an issue for the user if the device needs hardware repair or technical support (*Note: a device can be restored and therefore it is not a major issue, unless hardware damage otherwise covered by the warranty prevents restoration*).

What controls can be used to protect against it?

Before an organization chooses to implement a mobile solution in their environment they should conduct a thorough risk assessment. This risk assessment should include an evaluation of the dangers posed by jailbroken or rooted devices, which are inherently more vulnerable to malicious applications or vulnerabilities such as those listed in the [OWASP Mobile Security Top Ten Risks](https://www.owasp.org/index.php/OWASP_Mobile_Security_Project). Once this has assessment has been completed, management can determine which risks to accept and which risks will require additional controls to mitigate. Below are a few examples of both technical and non-technical controls that an organization may use.

**Technical Controls:**

Some of the detective controls to monitor for jailbroken or rooted devices include:

1. Identify 3rd party app stores (e.g., Cydia).
2. Attempt to identify modified kernels by comparing certain system files that the application would have access to on a non jailbroken device to known good file hashes. This technique can serve as a good starting point for detection.
3. Attempt to write a file outside of the application’s root directory. The attempt should fail for non-jailbroken devices.

Note: Most Mobile Device Management (MDM) solutions can perform these checks but require an application to be installed on the device.

**Non-Technical Controls:**

Organizations must understand the following key points when thinking about mobile security:

1. Perform a risk assessment to determine risks associated with mobile device use are appropriately identified, prioritized and mitigated to reduce or manage risk at levels acceptable to management.
2. Review application inventory listing on frequent basis to identify applications posing significant risk to the mobility environment.
3. Technology solutions such as Mobile Device Management (MDM) or Mobile Application Management (MAM) should be only one part of the overall security strategy. High level considerations include:
   1. Policies and procedures.
   2. User awareness and user buy-in.
   3. Technical controls and platforms.
   4. Auditing, logging, and monitoring.
4. While many organizations choose a Bring Your Own Device (BYOD) strategy, the risks and benefits need to be considered and addressed before such a strategy is put in place. For example, the organization may consider developing a support plan for the various devices and operating systems that could be introduced to the environment. Many organizations struggle with this since there are such a wide variety of devices, particularly Android devices.
5. There is not a ‘one size fits all’ solution to mobile security. Different levels of security controls should be employed based on the sensitivity of data that is collected, stored, or processed on a mobile device or through a mobile application.
6. User awareness and user buy-in are key. For consumers or customers, this could be a focus on privacy and how Personally Identifiable Information (PII) is handled. For employees, this could be a focus on Acceptable Use Agreements (AUA) as well as privacy for personal devices.

Conclusion  
Jailbreaking and rooting tools, resources and processes are constantly updated and have made the process easier than ever for end-users. Many users are lured to jailbreak or root their device in order to gain more control over the device, upgrade their operating systems or install packages normally unavailable through standard channels. While having these options may allow the user to utilize the device more effectively, many users do not understand that jailbreaking or rooting can potentially allow malware to bypass many of the device's built in security features. The balance of user experience versus corporate security needs to be carefully considered since all mobile platforms have seen an increase in malware attacks over the past year. Mobile devices now hold more personal and corporate data than ever before and have become a very appealing target for attackers. Overall, the best defense for an enterprise is to build an overarching mobile strategy that accounts for technical controls, non technical controls and the people in the environment. Considerations need to not only focus on solutions such as MDM, but also policies and procedures around common issues of BYOD, and user security awareness.