

# Jailbreaking, Rooting and adb



# News

---

- <https://www.spamhaus.com/resource-center/a-surge-of-malvertising-across-google-ads-is-distributing-dangerous-malware/>

# Getting apks Review

---

- Downloading apps from Androzoo
  - From browser on VM or **Android Studio Emulator**:
  - [https://androzoo.uni.lu/api/download?apikey=\\${APIKEY}&sha256=\\${SHA256}](https://androzoo.uni.lu/api/download?apikey=${APIKEY}&sha256=${SHA256})
  - **APIKEY** =  
97e238cdcee2ae34a73a81ee6d9c494e137ab6bf1a574623386a  
9a7256ca35f5
  - **SHA256** = first column in spreadsheet
  - Replace everything in red with actual keys

# Overview

---

- Explain the reasons for rooting or jailbreaking a mobile device, along with the potential security consequences
- Describe the process and tools for jailbreaking an iOS device
- Describe the process and tools for rooting an Android device
- Adb command reference

# Need for Jailbreaking and Rooting

---

- All popular devices have restrictions:
  - Permitted application install sources
  - Local device access privileges
  - Code signing requirements
- Administrators require access for:
  - Application binary collection for analysis
  - Runtime analysis of software
  - Filesystem monitoring and profiling
  - Penetration testing targets
- Enterprise mobile administrators should have at least one unrestricted device for each supported platform



# Unrestricted Device Legality

---

- In 2009, Apple pursued an injunction against developers who were publishing Apple iOS jailbreak tools, citing jailbreaking as a violation of the U.S. Digital Millennium Copyright Act (DMCA) and a copyright violation.
- In response, the U.S. Copyright Office published an exemption permitting device jailbreaking and removing the software restrictions on a phone that prevents it from being used with other carrier networks (<http://www.copyright.gov/1201/>).



# Unrestricted Device Warnings and Recommendations

---

## Warnings

- Rooted devices bypass secure boot and app execution and weaken malware protection
- In order to escape controls, it is necessary to run third-party tools
  - You don't know the quality of the tools
- Apple has indicated that jailbreaking will void the warranty of products
- Possibility of irrevocably "bricking" devices
- Opportunity to run software that might degrade performance or compromise device
- May be a EULA violation, regardless of legality

## Recommendations

- Don't jailbreak/unlock/root your everyday production devices
- Do jailbreak/unlock/root secondary devices for new access opportunities
- Check your organization's legal posture on permitted activities with corporate devices



# Jailbreaking vs Rooting

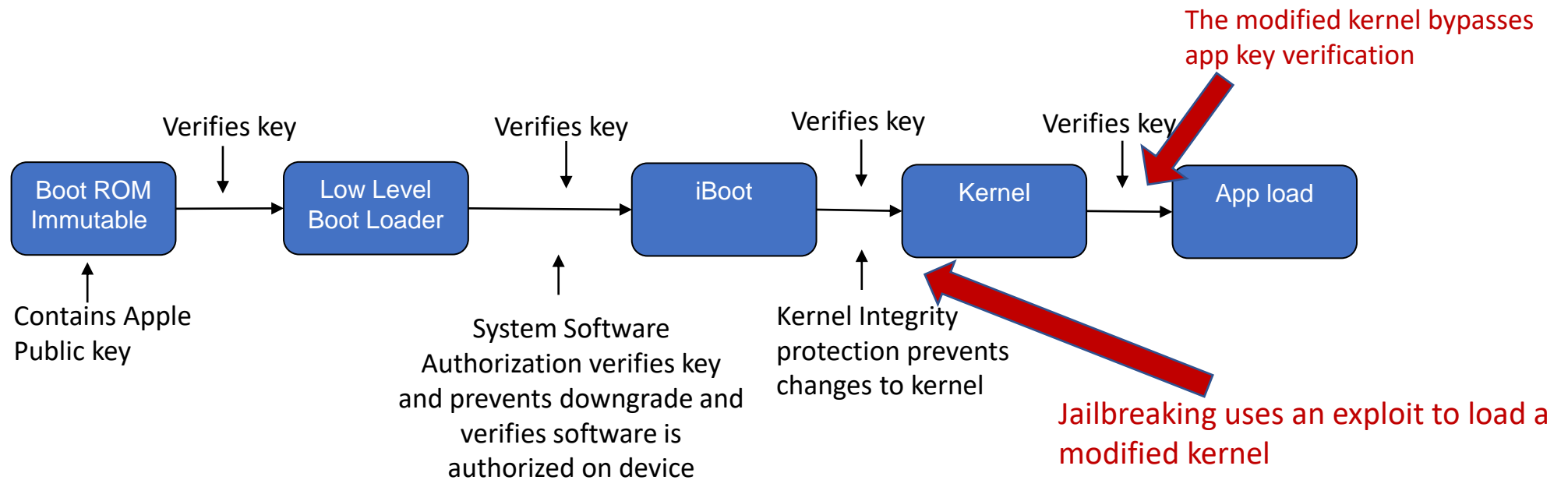
---

- Both seek to gain superuser access
- Jailbreaking allows iOS devices to load apps outside of the App Store
- Android users can sideload without rooting through a setting



# Jailbreaking Nullifies Signature Verification Process

- Boot process loads a modified kernel to bypass signature process
- Any OS update typically nullifies the jailbreak



# Jailbreaking iOS

---

- Types of jailbreak
  - Untethered – Persistent jailbreak. Much more difficult to accomplish as it requires modification of the signed kernel
  - Tethered – Temporary jailbreak for a single boot. Changes made while jailbroken will persist, but root access will not. Must be attached to computer to reboot to jailbreak mode, and may not reboot without computer
  - Semi-tethered - The device can start up on its own, but it will no longer have a patched kernel. It will still be usable for normal functions. To start with a patched kernel, the user must start the device with the help of the jailbreak tool
  - Semi-untethered jailbreak - The device can start on its own. On first boot, the device will not be running a patched kernel. However, rather than having to run a tool from a computer to apply the kernel patches, the user is able to re-jailbreak their device with the help of an app (usually sideloaded using Cydia Impactor) running on their device

# Jailbreaking Process Overview

---

- Break the signature verification process at the lowest level possible
  - Often through a buffer overflow
- Install a modified kernel that does not use the signature process
- Remount the root file system as readable and writeable
- Register a new service to provide read/write to the entire file system with root privileges
- Install various utilities and applications associated with the jailbreak



# Jailbreaking Tools

---

- Requires exploiting a vulnerability on the device
  - [https://theapplewiki.com/wiki/Jailbreak\\_Exploits](https://theapplewiki.com/wiki/Jailbreak_Exploits)
- Different tools for different devices and iOS versions
  - <https://theapplewiki.com/wiki/Jailbreak>
- Jailbreaking tools use iPhone Archive (IPA) files or progressive web apps
- Unc0ver and Electra (Chimera)
  - Download and install using Ignition or Tweakbox site/app
  - Run untethered jailbreak on older iOS
- Checkra1n
  - Semi-tethered jailbreak
- Use Cydia app store for questionable apps to test

# Rooting an Android Device



# Android Rooting

---

- Android users are not restricted to the official app store
  - Can allow unknown sources for apps
- Primary motivation is root access to file system
  - For testing and tweaking



# Android Rooting

---

- System root
  - Complete replacement of the kernel. Allowed by some devices
- Systemless root
  - Gains access through other exploits without modifying the system partition
- Different versions of Android and different devices have different vulnerabilities
- Rooting requires
  - Unlocking the boot loader (Some manufacturers provide instructions)
  - Researching the root exploit opportunities available for your device



# Android “Systemless” Root

---

- **Rooting with Magisk**
- Magisk ("Magic Mask") is one way to root your Android device. Its specialty lies in the way the modifications on the system are performed. While other rooting tools alter the actual data on the system partition, Magisk does not (which is called "systemless").
- <https://topjohnwu.github.io/Magisk/>

# Android Emulator and Root Access



# Android Studio Images and Root Access

---

- Google Services and Play Store
  - Play Store Access and app
- Google Services
  - Access to Play Store functions, but no Play Store App
- Open-source project (without Google API or Play Store)
  - No Play Store Access, but root file system access on all images
  - May affect app performance if it detects root

# adb Path

---

- Windows
  - Search adb.exe and right click to open folder
  - At top of folder right click and copy address as text
  - Search “env” and open Edit System Environment variables and select Environment variables
  - Edit path and add copied text, deleting “C:”
  - **Or run CMD as administrator and enter:**
    - `setx /M PATH "%PATH%;%userprofile%\appdata\local\android\sdk\platform-tools"`
- Mac
  - `export PATH=$PATH:~/Library/Android/sdk/platform-tool`



# Basic adb Commands

---

- Adb devices – shows connected devices
- Adb install “appname” – installs app on emulator or device
- Adb pull – pulls files and apks from device to user home directory
- Adb shell – connects to Linux shell on Android device
  - All commands are run on the emulator
  - su command to gain root if available
- Adb shell pm list packages – package manager to list installed apps
- adb shell pm path com.example.someapp – find path to apk file to pull
  - Named base.apk and should be renamed after pull

# In Class Quiz Question 1

---

- Create and start Android emulator with Pixel 3 Play Store and API 30 with Play Store
- Adb shell to device
  - attempt root access with su
  - ps – to list processes – note last user name for quiz
- Download Uncrackable1 from Canvas and install with adb
- Open app and note it does not detect root
- **Close tab to shutdown emulator**

# In Class Quiz Question 2

---

- Create and start Android emulator with Pixel 3XL **no** Play Store and API 30 Open Source – from x86 images
- Adb shell to device
  - attempt root access with su
  - ps – to list processes – note last user name for quiz
- Install Uncrackable1 with adb
- Open app and note it **does** detect root

# Summary

---

- Reasons for gaining unrestricted access
- Cautions about gaining unrestricted access
- Jailbreaking process and tools
- Rooting process and tools

