Rockchip
瑞芯微电子

Magisk installation guidelines

release version: 1.0.1

Date: 2019.12

Preface

Overview

This document shows how to install Magisk to Rockchip Android Pie devices and Android Q devices.

Intended audience

This document is suitable for the following engineers:
Software Developer

Revision record

DATE	Revision	Author	Reviewer	Modify description
2019-01-10	V1.0.0	xjq	cw	Release
2019-12-03	V1.0.1	xjq	cw	Support Android Q

Contents

Introduction

Magisk is a suite of open source tools for customizing Android, supporting devices higher than Android 5.0 (API 21). It covers the fundamental parts for Android customization: root, boot scripts, SELinux patches, AVB2.0 / dm-verity / forceencrypt removals etc.

Furthermore, Magisk provides a **Systemless Interface** to alter the system (or vendor) arbitrarily while the actual partitions stay completely intact. With its systemless nature along with several other hacks, Magisk can hide modifications from nearly any system integrity verifications used in banking apps, corporation monitoring apps, game cheat detections, and most importantly Google's SafetyNet API.

Reference: https://github.com/topjohnwu/Magisk

Installation Methods

There are two ways to integrate Magisk

- Flash boot image which is patched by Magisk Manager
- Flash a Magisk zip via third-party recovery（e.g. TWRP）

Flash boot image which is patched by Magisk Manager

Most rockchip devices are non-A/B. In Android Pie, non-A/B devices should be system-as-root, which don't contain ramdisk in boot.img. Since Magisk must install Magisk files into the ramdisk in boot image, we have to ensure that boot.img contains ramdisk. In Android Q, ramdisk come back to boot.img.

According to A/B devices, Ramdisk can be added to boot.img, then set skip_initramfs in cmdline to skip ramdisk in boot.img.

Preparation

- Build boot.img(must contain ramdisk)

Enable BOOTIMG_SUPPORT_MAGISK, and build boot.img.

```
device/rockchip/rk3328/rk3328_box BoardConfig.mk
BOOTIMG_SUPPORT_MAGISK := true

make bootimage
```

NOTE: In Android Q, please skip this step since boot.img contains ramdisk already.

- Download MagiskManager:

https://github.com/topjohnwu/Magisk/releases

Installation step

- Push boot.img to device

```
adb push path/to/boot.img /storage/emulated/0/Download
```

- Install MagiskManager

```
adb install /path/to/MagiskManager.apk
```

- Patch boot.img

Patch boot.img by MagiskManager

  - Click install->Patch Boot Image File, select the boot.img, which has been pushed in the first step.
  - MagiskManager will download Magisk zip and patch boot.img automatically.

- Pull boot_patched.img to your PC

```
adb pull /storage/emulated/0/Download/boot_patched.img
```

- Flash boot_patched.img to boot partition by AndroidTools

NOTE: When clicking Patch Boot Image File, the apk may crash, which should be solved by installing third-party file explorer (e.g. ES File Explorer File Manager).

Please disable Magisk Hide if **adb is not available** after you install magisk successfully.

  - click Settings-> Settings-> Magisk Hide
  - reboot device

Flash Magisk zip via TWRP

Preparation

- Build TWRP：https://github.com/rockchip-software/TWRP
- Build boot.img with ramdisk(refer to sections above)
- Download Magisk.zip：https://github.com/topjohnwu/Magisk/releases

Installation step

- Flash TWRP(recovery.img) and boot.img by AndroidTool
- Push Magisk.zip to device

```
adb push /path/to/Magisk.zip /sdcard/
```

- Reboot to recovery

```
adb reboot recovery
```

- Install Magisk.zip via TWRP
- Reboot device
- Install MagiskManager

```
adb install /path/to/MagiskManager.apk
```

SafetyNet Check

There are two parts to a SafetyNet check, CTS compatibility and Basic integrity. The CTS check is a server side checkup that's difficult to spoof, while Basic integrity is done on the device side and is a lower level of security. Some apps only use the Basic integrity part of the SafetyNet API and thus can be used even if SafetyNet doesn't fully pass.

If you can't pass SafetyNet, but Basic integrity shows as true, that basically means Google doesn't trust your device for some reason. You should be able to fix this by

matching prop values with a ROM that passes SafetyNet. Try changing your device's ro.build.fingerprint to a device's/ROM's that is known to pass SafetyNet. The Magisk module MagiskHide Props Config can do this.

Install through the Magisk Manager Downloads section. Or, download the zip from the Manager or the module support thread, and install through the Magisk Manager -> Modules, or from recovery(e.g. TWRP). After installing the module and rebooting, run the command:

```
adb shell
su
props
```

Picked a certified fingerprint from the provided list.

More information: https://www.didgeridoohan.com/magisk/MagiskHideSafetyNet
https://github.com/Magisk-Modules-Repo/MagiskHidePropsConf

NOTE: if you have integrated other root tools, please remove them first.