

Android

- [SSH for Android](#)
- [Remote screen](#)
- [ROMs](#)
 - [Redmi Note 8](#)
 - [Sauvegarde des cellulaires](#)
 - [ADB Configuration script](#)
 - [Play Integrity/SafetyNet](#)
- [USB Debugging authorization](#)
- [HK1 RBOX X4](#)
- [Resources](#)
- [Advanced Charging Controller \(ACC\)](#)
- [ADB Commands](#)

SSH for Android

SSH on Android

Step by step

- Install F-Droid app from Internet
- Install Termux app from F-Droid
- Install openssh binary from Termux
 - Open a Termux window
 - Run the following

```
pkg upgrade  
pkg install openssh
```

Usage

- Start the SSH server
(Default SSH port in Termux is 8022)

```
sshd
```

- Stop the SSH server

```
pkill sshd
```

Configuration ADB/SCRCOPY via wireless

Android device v10+ must be paired first to work by wireless.

- Settings/Developer Options/Wireless debugging
- Choose "Pair device with pairing code"
- Use displayed information to perform pairing on remote machine

```
adb pair {android ip address}:{android ip port}
```

<https://developer.android.com/tools/adb#connect-to-a-device-over-wi-fi-android-11+>

Setup public key authentication

- Copy public key

Use scp to copy file locally then add pubkey content to authorized_keys

```
cat /sdcard/Download/id_rsa_rasp.pub >>
/data/data/com.termux/files/home/.ssh/authorized_keys
```

or

Configure temporary password then copy using `ssh-copy-id` command:

```
whoami
passwd
ssh-copy-id -p 8022 -i {id_rsa file} {user}@{android ip address}
```

- Allow storage access to Termux

```
termux-setup-storage
```

- Disable password auth in sshd configuration file

Set `PasswordAuthentication` to **no** in `$PREFIX/etc/ssh/sshd_config`

```
sed -Ei"_ori" 's/[#]?PasswordAuthentication yes|PasswordAuthentication no|g'
$PREFIX/etc/ssh/sshd_config
```

- Restart service

```
kill sshd; sshd
```

Resources

<https://github.com/termux>

https://wiki.termux.com/wiki/Remote_Access

https://wiki.termux.com/wiki/Main_Page <https://joeprevite.com/ssh-termux-from-computer/>

<https://github.com/dmdhrumilmistry/Termux-SSH>

Remote screen

<https://github.com/Genymobile/scrcpy>

```
# Drag and drop files in the scrcpy window (default is /sdcard/Download)
scrcpy --tcpip=192.168.66.178 --push-target=/sdcard/Documents/

# Window scaled to local optimized (no audio, etc)
scrcpy --tcpip=192.168.66.178 --no-audio

# Full optimized cmdline
scrcpy --tcpip=androidtv-01 --no-audio --max-size=1024 --push-target=/sdcard/Documents/
```

Avoid the nagging USB authorization screen

```
# requires root
- Go to C:\Users<profile_name>.android\adbkey.pub
- Copy the file to any location in your phone using USB cable.
- Using Root Explorer, move the adbkey.pub onto this location: /data/misc/adb/
- rename it from adbkey.pub to adb_keys and allow read and write permissions
- Reboot the device.
```

Keyboard

Switch fullscreen mode MOD+f

Rotate display left MOD+← (left)

Rotate display right MOD+→ (right)

Resize window to 1:1 (pixel-perfect) MOD+g

Resize window to remove black borders MOD+w | Double-left-click¹

Click on HOME MOD+h | Middle-click

Click on BACK MOD+b | Right-click²

Click on APP_SWITCH MOD+s | 4th-click³

Click on MENU (unlock screen)⁴ MOD+m

Click on VOLUME_UP MOD+↑ (up)
Click on VOLUME_DOWN MOD+↓ (down)
Click on POWER MOD+p
Power on Right-click²
Turn device screen off (keep mirroring) MOD+o
Turn device screen on MOD+Shift+o
Rotate device screen MOD+r
Expand notification panel MOD+n | 5th-click³
Expand settings panel MOD+n+n | Double-5th-click³
Collapse panels MOD+Shift+n
Copy to clipboard⁵ MOD+c
Cut to clipboard⁵ MOD+x
Synchronize clipboards and paste⁵ MOD+v
Inject computer clipboard text MOD+Shift+v
Enable/disable FPS counter (on stdout) MOD+i
Pinch-to-zoom Ctrl+click-and-move
Drag & drop APK file Install APK from computer
Drag & drop non-APK file Push file to device

Copy-paste

Any time the Android clipboard changes, it is automatically synchronized to the computer clipboard.

Any Ctrl shortcut is forwarded to the device. In particular:

```
Ctrl+c typically copies  
Ctrl+x typically cuts  
Ctrl+v typically pastes (after computer-to-device clipboard synchronization)
```

This typically works as you expect.

The actual behavior depends on the active application though. For example, Termux sends SIGINT on Ctrl+c instead, and K-9 Mail composes a new message.

To copy, cut and paste in such cases (but only supported on Android ≥ 7):

```
MOD+c injects COPY  
MOD+x injects CUT  
MOD+v injects PASTE (after computer-to-device clipboard synchronization)
```

In addition, MOD+Shift+v injects the computer clipboard text as a sequence of key events. This is useful when the component does not accept text pasting (for example in Termux), but it can break non-ASCII content.

WARNING: Pasting the computer clipboard to the device (either via Ctrl+v or MOD+v) copies the content into the Android clipboard. As a consequence, any Android application could read its content. You should avoid pasting sensitive content (like passwords) that way.

Some Android devices do not behave as expected when setting the device clipboard programmatically. An option `--legacy-paste` is provided to change the behavior of Ctrl+v and MOD+v so that they also inject the computer clipboard text as a sequence of key events (the same way as MOD+Shift+v).

To disable automatic clipboard synchronization, use `--no-clipboard-autosync`

ROMs

ROMs

Redmi Note 8

Codename: ginkgo

To find product name : fastboot getvar product

Telegram groups:

t.me/rn8discussions

t.me/RedmiNote8Official

Forum XDA

<https://forum.xda-developers.com/redmi-note-8>

TWRP

<https://forum.xda-developers.com/redmi-note-8/how-to/guide-unlocking-bootloader-twrp-rooting-t4031831>

- TWRP Official :

<https://forum.xda-developers.com/t/recovery-official-twrp-teamwin-recovery-redmi-note-8.4301393/>

Unlocking phone

<https://forum.xda-developers.com/t/guide-all-about-unlocking-bootloader-twrp-and-rooting-redmi-note-8-8t.4031831/>

USB Debugging (with security)

This requires root, shell access with su, and a way to edit text files as root. I used Termux and an android root file manager that can edit text files.

First, run the following command (su first, to switch to root):

```
setprop persist.security.adbinstall 1
```

If you also want to enable "USB debugging (Security settings)", for granting permissions, simulating UI input, etc. via ADB, run the following command too:

```
setprop persist.security.adbinput 1
```

Next, we need to edit the edit the file

```
/data/data/com.miui.securitycenter/shared_prefs/remote_provider_preferences.xml
```

so that MIUI actually lets us do these things:

Add the following line to the XML file (with all the other ones, before the closing tag):

```
<boolean name="security_adb_install_enable" value="true" />
```

TWRP

Was in TWRP 3.3.1-9

To install the correct TWRP version, check the "base" of the TWRP indicated after the version number. E.G. :

`twrp-3.6.2_9-0-ginkgo.img` is a base Android 9-0

`twrp-3.5.2_10-2-ginkgo.img` is a base Android 10-2

`twrp-3.7.0_12-1-ginkgo.img` is a base Android 12-1

Some Android versions cannot decrypt your data folder because of incompatibility issues. Simply downgrade to a lower Android base TWRP.

- Enter Fastboot to install latest TWRP

```
# To check current recovery tool (with phone powered up)
adb reboot recovery

# Boot to recovery
adb reboot bootloader
```

```
# Confirm correctly in fastboot
fastboot devices
# Install latest TWRP
cd "/home/wes/Documents/Cellulaires-ROMs/Redmi Note 8"
fastboot flash recovery twrp-3.7.0_12-1-ginkgo.img
# Shutdown phone then "Volume Up + Power" to enter recovery
fastboot reboot
```

First time installation

You might need to use Lineage OS recovery the first time to initialize a factory reset. Simply follow the instruction here : <https://wiki.lineageos.org/devices/ginkgo/install/variant1#installing-a-custom-recovery-using-fastboot>

Sideload the custom ROM rather than Lineage.

ArrowOS

- Boot in recovery
- Wipe data, cache, dalvik cache
- Flash latest android-11 firmware (Ginkgo | Willow)

```
adb push Arrow-v13.1-ginkgo-OFFICIAL-20230829-GAPPS.zip /sdcard
```

- Flash ROM zip
- Format data (if encrypted)
- Reboot
- Configure system manually
- Install Magisk for root

```
adb reboot recovery
adb sideload Magisk.v26.3.apk
```

No need to follow this, simply install by sideloading + app within OS

Since Redmi Note 8 has no ramdisk, a special guide is required.

- Install the Magisk app from the official git repo.
- Push installed recovery locally : `adb push twrp-3.7.0_12-1-ginkgo.img /sdcard/Documents/`
- Execute Magisk app
- Press the Install button on the Magisk card.

- Enable "Recovery Mode" under Options
- Choose "Select and Patch a File" under Method, and select the local recovery image.
- Retrieve the patched image locally : adb pull
/sdcard/Download/magisk_patched_[random_strings].img
- Reboot in fastboot and flash recovery : fastboot flash recovery
/path/to/magisk_patched.img
- Configure Magisk
 - Settings/Systemless hosts
 - Settings/Zygisk
- Install Magisk Modules

microG installer : https://github.com/nift4/microg_installer_revived

safetynet fix : <https://github.com/kdrag0n/safetynet-fix>

```
# microG
adb push microG_Installer_Revived.zip /sdcard/Documents/
# Google safetynet fix
adb push safetynet-fix-v2.4.0.zip /sdcard/Documents/
```

- Install EasySSHFS
<https://github.com/bobrofon/easysshfs/releases>
adb push easysshfs-0.5.10-release-signed.apk /sdcard/Documents/
- Install neo-backup
<https://github.com/NeoApplications/Neo-Backup/releases> adb push
Neo_Backup_release_832.apk /sdcard/Documents/

Reconfigure some apps

Some apps require to re-login after restoration :

- Banks (Wise ; Desjardins)
- eMails (Proton ; Tutanota ; etc)
- Social apps (Instagram ; Viber ; Signal ; Telegram ; etc)
- Music (Deezer)
- xMatters

Some interesting modules

<https://github.com/Magisk-Modules-Repo/ssh>

Lineage OS

<https://wiki.lineageos.org/devices/ginkgo/variant1/>

Boot to Fastboot/bootloader

```
adb reboot bootloader
```

Flash vbmeta file (to disable boot security feature)

```
fastboot --disable-verity --disable-verification flash vbmeta vbmeta.img
```

Flash and boot to recovery TWRP

```
fastboot flash recovery twrp-3.3.1-9-ginkgo-mauronofrio.img  
fastboot boot twrp-3.3.1-9-ginkgo-mauronofrio.img
```

Format Data

```
TWRP > Wipe > Format Data (not in advanced mode) - No reboot to system
```

Reboot in recovery

```
TWRP > Reboot > Recovery
```

Wipe again

```
TWRP > Wipe > Advanced > Cache, data, dalvik, system
```

Flash Disable_DM-Verity_ForceEncrypt

```
https://forum.xda-developers.com/android/software/universal-dm-verity-forceencrypt-t3817389/
```

Flash ROM

Flash Magisk

Reboot to ROM

Reboot to recovery

Generate and Install Smali Patcher module in Magisk

Flash nanodroid setup (to configure nanodroid) - doesn't works.

Flash nanodroid

Source:

<https://forum.xda-developers.com/showpost.php?p=81403899&postcount=37>

Install smali patcher for signature spoofing

<https://forum.xda-developers.com/apps/magisk/module-smali-patcher-0-7-t3680053>

Baseband

Version

Firmware	Baseband (BAR_GEN_PACK)
V11.0.8.0.PCOMIXM	1.252721.1.260983.1
V11.0.7.0.PCOMIXM	1.246073.2.252721.1
20.1.16 (China)	1.241207.1.242935.1
V11.0.6.0.PCOMIXM	1.239860.1.241207.1
V11.0.3.0.PCOMIXM	1.223797.1.225109.3

Sauvegarde des cellulaires

Sauvegarde cellulaires

<https://sebsauvage.net/wiki/doku.php?id=backup-telephone>

<https://alexandre.deverteuil.net/post/synchronizing-files-across-phone-laptop-nas-with-nextcloud-and-syncthing/>

- SMS : <https://play.google.com/store/apps/details?id=com.idea.backup.smscontacts>
- Apps : <https://f-droid.org/en/packages/com.machiav3lli.backup/>
- Accès au systèmes de fichiers :
 - EasySSHFS : <https://f-droid.org/en/packages/ru.nsu.bobrofon.easysshfs/>
 - primitive ftpd : <https://f-droid.org/fr/packages/org.primftpd/>

Rapport

Valider le contenu du stockage:

```
# Vue globale du stockage
df -h
cd /data/
du -m | awk '$1 >= 100' | sort -g # List directories above 100 MB
du -am -d1 | awk '$1 >= 100' | sort -g # List all above 100 MB at depth level 1
du -am -d2 | awk '$1 >= 100' | sort -g # List all above 100 MB at depth level 2
```

Étapes

```
# Pré-requis - récup du cell id
adb devices -l
```

1. Backup des apps avec neobackups
2. Backup Signal
Settings/Chats/Chats backup/Create backup
(passphrase in keepass)

```
adb -s <cellulaire id> pull -a /sdcard/Documents/Backup_signal ~/Documents/sauvegarde
```

3. Backup Markor

```
adb -s <cellulaire id> pull -a /sdcard/Documents/markor ~/Documents/sauvegarde
```

4. Sauvegarde des settings

```
mkdir /sdcard/Documents/backup_settings  
su - # passe en root  
cp /data/system/users/0/* /sdcard/Documents/backup_settings/
```

5. Copie local de la sauvegarde

```
adb -s <cellulaire id> pull -a /sdcard/backup_neobackup ~/Documents/sauvegarde
```

6. Copie de la sauvegarde vers le nouveau cell

```
adb -s <cellulaire id> push ~/Documents/sauvegarde/backup_neobackup /sdcard/
```

7. Installation des apps minimums

```
adb -s <cellulaire id> install ~/Documents/Cellulaires-ROMs/magisk-v28.apk  
adb -s <cellulaire id> install ~/Documents/Cellulaires-ROMs/droid-ify-v0.6.3.apk  
adb -s <cellulaire id> install ~/Documents/Cellulaires-ROMs/auroraStore-4.6.2.apk  
adb -s <cellulaire id> install ~/Documents/Cellulaires-ROMs/neo_Backup_838_release.apk
```

8. Restauration des apps via Neobackup
9. Configuration des connexions pour les apps
 - Signal
 - Viber
 - Whatsapp
 - Facebook (Messenger)
 - Desjardins
 - Wise
 - Picard
 - Proton Mail
 - Proton VPN
 - Tuta
 - Manuvie
 - ...etc...

Configuration EasySSHFS

wes

192.168.66.103

remote path : ~/Documents/sauvegarde

(create folder in Android home named "sauvegarde")

Ressources

<https://github.com/newhinton/Round-Sync>

Outil pour sync 1-direction basé sur rclone

ADB Configuration script

ADB Config

Check logs using logcat

- Find name of app
- filter with the name

```
$ pm list packages | grep -i seed
package: com.stevesoltys.seedvault
$ logcat -e seedvault
# or use a simple grep
logcat | grep 'vault'
```

```
# This is specific to SeedVault (https://github.com/seedvault-app/seedvault/blob/android15/logcat-verbose.sh)
set -ex

adb shell setprop log.tag.BackupManagerService VERBOSE
adb shell setprop log.tag.BackupManagerConstants VERBOSE
adb shell setprop log.tag.BackupTransportManager VERBOSE
adb shell setprop log.tag.KeyValueBackupJob VERBOSE
adb shell setprop log.tag.KeyValueBackupTask VERBOSE
adb shell setprop log.tag.TransportClient VERBOSE
adb shell setprop log.tag.BackupAgent VERBOSE
adb shell setprop log.tag.PMBA VERBOSE
```

Navigate through the slots

To flash permanently, first switch/select any slot you want or it will **automatically flashed to the active slot**.

```
# Check current active slot
adb shell getprop ro.boot.slot_suffix
fastboot getvar current-slot

# Switching slots
fastboot set_active other //switch to inactive slot
fastboot set_active (a or b) //switch to specified slot
```

Check current slot and baseband version

```
adb shell getprop | grep -Ei '(gsm.*baseband|slot_suffix)'
```

Have both slots (A/B) with same verity/verification configuration

```
fastboot flash vbmeta --disable-verity --disable-verification --slot=all vbmeta.img
```

Secure phone (to review)

```
#!/bin/bash
adb root
adb remount
adb pull /system/build.prop .
sed -i "s/ro.secure=0/ro.secure=1/" build.prop
sed -i "s/ro.debuggable=1/ro.debuggable=0/" build.prop
adb push build.prop /system/build.prop
rm build.prop
```


ROMs

Play Integrity/SafetyNet

Play Integrity

- ROM dummies (for vendor data file) :

<https://dumps.tadiphone.dev/dumps>

USB Debugging authorization

Copy public key

- Windows

```
adb push C:\Users\wesle\.android\adbkey.pub /data/misc/adb/adb_keys
```

- Linux

```
adb push ~/.android/adbkey.pub /data/misc/adb/adb_keys
```

HK1 RBOX X4

HK1RBOX-X4

Amlogic S905X4

Specifications (adb shell getprop)

```
ro.product.build.fingerprint]:  
[Amlogic/ah212/ah212:11/RQ3A.210805.001.A1/user12212324:user/release-  
keys]  
[ro.product.build.id]: [RQ3A.210805.001.A1]
```

GPU (adb shell dumpsys SurfaceFlinger | grep GLES) :

```
GLES: ARM, Mali-G31, OpenGL ES 3.2 v1.r25p1-01bet0-6a4488b89-  
0567174f3.3ce44ab07ffe09dde7ee2f5999c7dd83
```

Backup original ROM

<https://slimboxtv.ru/obnovlenie-proshivki-s-soxraneniem-dannyx-v-usb-burning-tool/>

Custom ROM

<https://slimboxtv.ru/transpeed-x4-x4s/>

Procedure:

- Start the Amlogic Flash tool with administrative privilege
- Load the custom image
- Leave the options by default (erase flash, etc)

- Press the `start` button
- Push `reset` button located inside the audio/jack connector
- Plug the USB to the PC The USB2 connector (not the blue one) is pretty tight so to avoid bad connection, put the box on the side having the USB port facing the ceiling. Don't move at all during the process.
- Wait for the flash to proceed

Configuration

- Update Magisk app and system using Magisk app
 - Configure systemless hosts file
 - Configure denylist
- Enable all slimbox features in the `internet` menu
- Install basic packages
 - AdAway
 - Amaze
 - F-Droid
 - Firefox
 - Neo Backup
 - NewPipe
 - Tivimate
 - VLC
 - WG Tunnel
 - Wireguard
- Update all packages
- Freeze APKPure package

```
# Find package name
adb shell pm list packages -e | grep -i apk

# Disable/freeze package
adb shell pm disable com.apkpure.aegon

# Stop all processes
adb shell ps -Aw | grep -i apkpure

# Grab the PID above and kill processes
kill ${PID}
```

- WireGuard Configuration
 - Add the below line in each conf file located in `/data/data/com.wireguard.android/files` : `IncludedApplications = ar.tvplayer.tv`
 - Configure the "Always-on VPN" settings in the "Network & Internet" settings
- Install Viper4Android <https://github.com/programminghoch10/ViPER4AndroidRepackaged>
Download latest package and install using Magisk.

Troubleshooting

- Check the current network connections

```
adb shell netstat -lpewt
```

Infos

<https://androidpctv.com/firmware-hk1-rbox-vontar-x4-transpeed-x4-rom/>

Resources

<https://en.aptoide.com/>

<https://apkpure.com/>

<https://modyolo.com/ott-navigator-iptv.html>

Advanced Charging Controller (ACC)

Link

<https://github.com/VR-25/acc>

<https://xdaforums.com/t/advanced-charging-controller-acc.3668427/>

Config

Configure day and night profiles (aka "Smart Charging")

```
#= Profile without charge cooldown
# Profil Jour
acc -c a ": day profile; at 6:00 \"acc -s rc=75 pc=80 mcc=750 mcv=4100; acc -n 'Switched to
day profile' \"
# Profil Nuit
acc -c a ": night profile; at 21:00 \"acc -s rc=45 pc=50 mcc=500 mcv=3900; acc -n 'Switched to
night profile' \"

#= Profile with charge cooldown
# Profil Jour
acc -c a ": day profile; at 6:00 \"acc -s rc=75 pc=80 mcc=750 mcv=4100 cch=50 cp=5; acc -n
'Switched to day profile' \"
# Profil Nuit
acc -c a ": night profile; at 21:00 \"acc -s rc=45 pc=50 mcc=500 mcv=3900 cch=50 cp=5; acc -n
'Switched to night profile' \"
```

Informations importantes

Ever wondered why lithium ion batteries aren't sold fully charged? They're usually ~40-60% charged. Why is that? Keeping a battery fully drained, almost fully drained or 70%+ charged for a long times, leads to significant (permanent) capacity loss

- Night/heavy-duty/forever-plugged profile: keep capacity within 40-60% (e.g., acc 50 45) and/or voltage around ~3900 mV
- Day/regular profile: max capacity: 75-80% and/or voltage no higher than 4100 mV
- Travel profile: capacity up to 95% and/or voltage no higher than 4200 mV

The maximum voltage actually defines the percentage of battery, 4200 mV ~95%, 4100 mV ~80%

Always Limit the Charging Current If Your Battery is Old and/or Tends to Discharge Too Fast

This extends the battery's lifespan and may even reduce its discharge rate.

- 750-1000mA is a good range for regular use.
- 500mA is a comfortable minimum - and also very compatible.

If your device does not support custom current limits, use a dedicated ("slow") power adapter.

Exemples

- `acc -s pc=60 mcc=500 mcv=3900`

This keeps battery capacity between 55-60%, limits charging current to 500 mA and voltage to 3900 millivolts. It's great for nighttime and "forever-plugged".

ADB Commands

Basic ADB Commands

```
# List connected devices
adb devices -l

# Run shell
adb shell

## Run "su" to elevate to root within the box
```