Android: All-In-One Solution - Custom ROM (GrapheneOS), Rooting, FOSS, NetHunter, Additional Configurations, MDM

This is my favourite project that I worked on for fun for myself because I love Android Phones (Google Pixels especially) and I've dug into this rabbit hole - I'm also trying to learn android app sec for bug hunting, RATs, etc in my spare time.

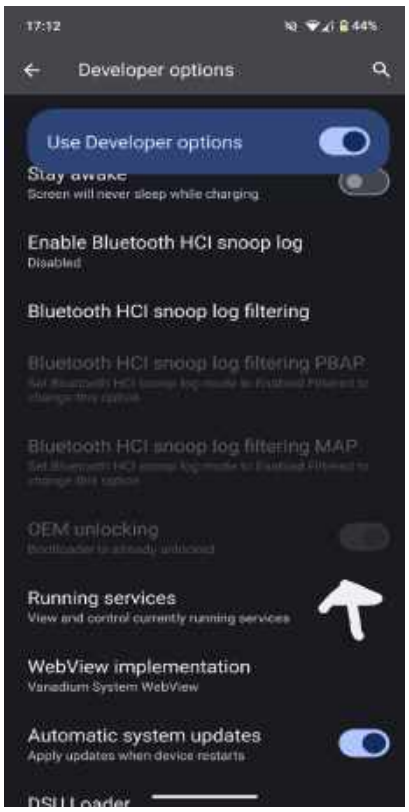
Requirements for all this to work: Google Pixel 7/a/pro or higher (64GB storage and atleast 8GB RAM), Windows 10 or 11/Linux (I will use Windows), USB-C Cable, SDK-Platform Tools Download

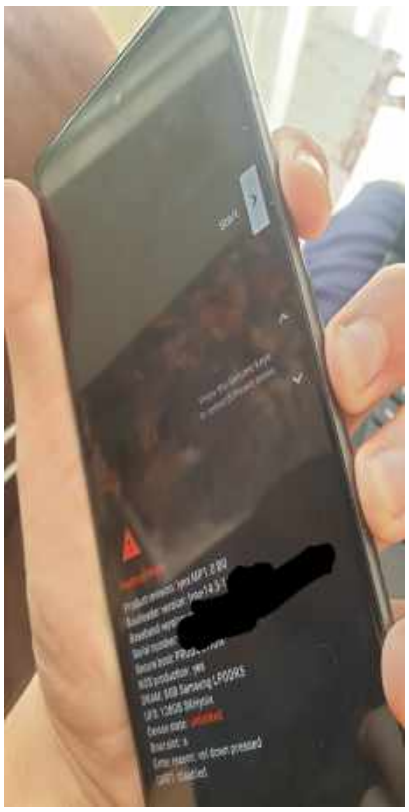
GrapheneOS is literally the same as normal OS but debloated and you can install all Play Store apps, APKs, including Banking Apps as usual, except Google Pay (NFC). Proceeding with Step 1 will factory reset your device and wipe your data.

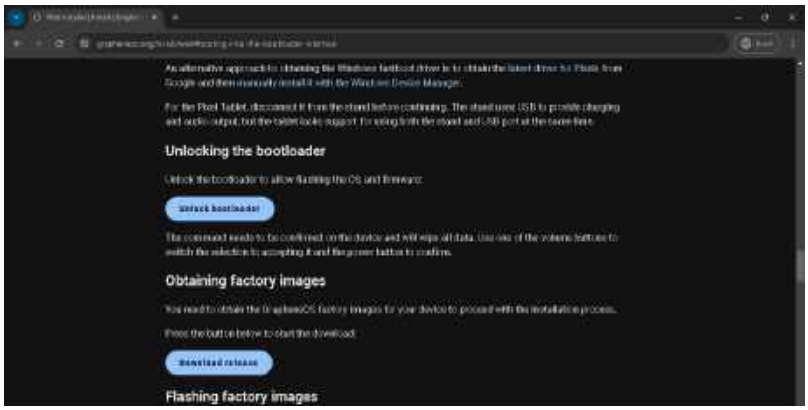
Step 1: Easily Install GrapheneOS through the web on the fly

1. Settings >> About Phone >> Scroll down and tap Build Number 7 times >> Go back >> System >> Developer Options >> Enable OEM Unlock (and optionally USB debugging).
2. Shutdown your Pixel 7/a/pro fully. >> Now hold the power and volume down button at the same time until you see **"Fastboot Mode"**.
3. Open this link in a new tab (Chrome) on Windows https://grapheneos.org/install/web#unlocking-the-bootloader.
4. Connect the phone to Windows with your USB cable. >> Allow any perms asked.
5. Go to windows update and check for updates >> advanced options >> optional updates and install any available.
6. Now go back to that link and easily follow the steps provided by GrapheneOS to install via the web, by first starting off unlocking the bootloader and confirming it.
7. Once Finished, Setup device as usual.

Few things to Note: We first need to enable OEM in settings so that we can access Fastboot which will allow us to unlock the bootloader and rewrite the existing OS (stock). We install any Windows updates and drivers so that the device and Windows recognise each other.







Step 2: Easily Rooting Graphene

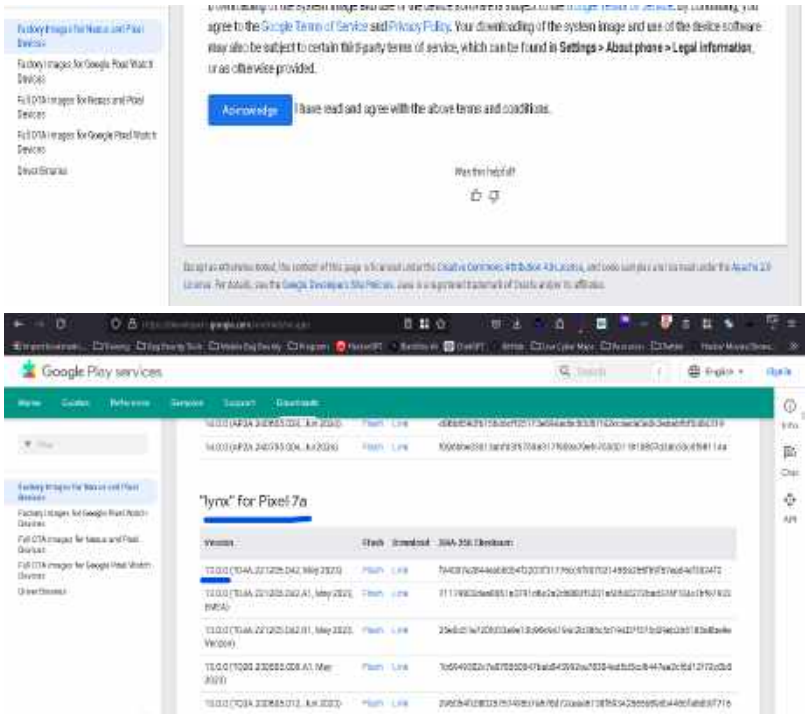
This step is optional if you want additional features, mods, customs and cool stuff. **This does compromises alot of GrapheneOS Security. (Risk: Apps requiring root access means it has high-level permissions. If exploited, it could access and modify sensitive data.)** This step will not wipe your data as long as bootloader is unlocked.

1. Visit this link https://developers.google.com/android/images >> Scroll down >> Acknowledge Terms of Service >> You'll see **"Factory Images for Nexus and Pixel Devices"**.
2. On your Phone >> Settings >> About Phone >> Click Android Version >> Take note of the **"Bootloader version"** (in my case, **"lynx"**(numbers) android 14).

3. On Windows, go back to the link >> Scroll down until you find your version (in my case, **""lynx" for Pixel 7a"**) >> Click download link and wait for it. (If you are android 13, download the latest version of **"13.0.0"**, if you are android 14, download the latest **"14.0.0"**)
4. Unzip (Extract) the factory image >> look for **"image-[device codename]-version.zip"** and extract that as well >> Copy/Transfer the **"init_boot.img"** onto your Pixel storage.
5. On your phone >> Download and install the latest version of magisk https://github.com/topjohnwu/Magisk/releases >> Run Magisk >> tap on the install button >> choose **"Select and Patch a File"** under Method >> select the **"init_boot.img"** that you copied into your Pixel >> Click **"Let's Go"** >> Once the file is patched, it should be named **"magisk_patched_xxx.img"** >> copy that file back to Windows >> rename it to something easier if you want.
6. Download SDK Platform tools On Windows https://dl.google.com/android/repository/platform-tools-latest-windows.zip (extract it) >> Open it >> Right click (more options) >> Open in terminal >> Now test and run `./adb devices` from terminal (You should see it run successfully with no devices attached) >> Leave the terminal open.
7. On your phone >> Enable USB debugging if you haven't done so in developer options >> Fully shutdown your Pixel >> Hold the volume down and power buttons together till you see **"fastboot"**.
8. Plug your phone into your computer using a USB cable >> From terminal, type `./fastboot devices`, you should see it connected >> Enter the command `./fastboot flash init_boot path/to/magisk_patched_xxx.img` (the path to where you copied **"magisk_patched_xxx.img"** on Windows).
9. Press the power button to start the phone >> open Magisk >> follow the steps, you'll be asked to reboot again.
10. To confirm you're root, go on magisk and verify you can click on superuser, if so, congrats :)

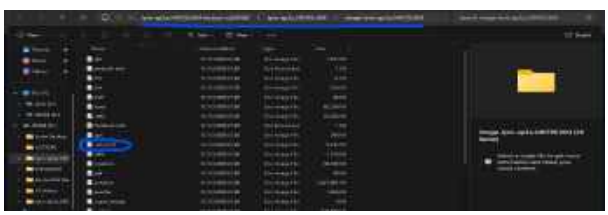
Few things to Note: We need to grab the exact version/model of our init boot image, which Magisk patches and creates its own boot file for root access, then using adb tools - we re-write the original boot image with patched Magisk. Another very important thing to note, after an OS update, the **"init_boot.image"** changes, so you'll have to repeat step 7-10 and re-write the **"magisk_patched_xxx.img"** so keep the magisk image saved. □


19:11
Android version
Android version
14
Android security update
5 July 2024
Baseband version
g5300g-240308-240517-B-11857467,g5300g-240308-240517-B-11857467
Bootloader version
lynx-14.5-11677884
Kernel version
5.10.219-android13-4-g14ccbcfedbeb
#1 Tue Jul 16 15:12:32 UTC 2024
Build number
20240716.00

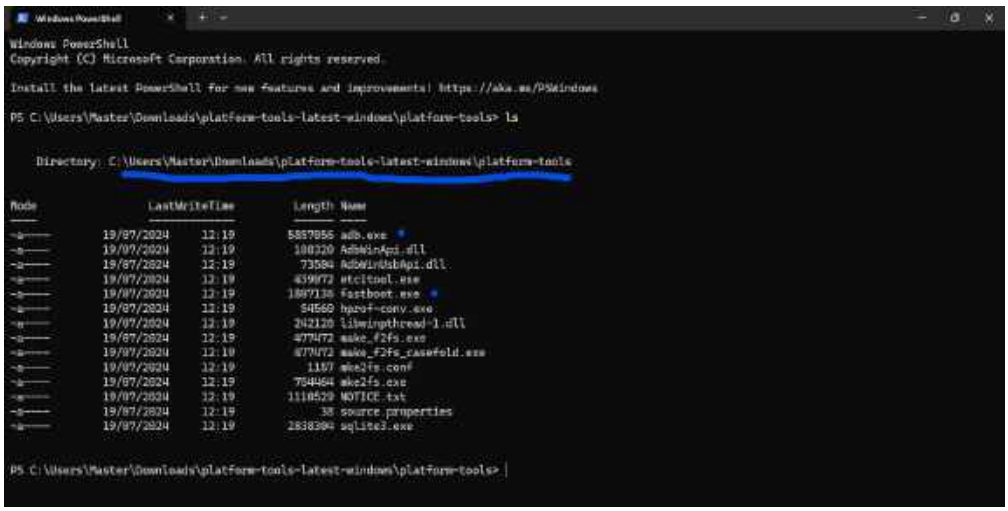

agree to the Google Terms of Service and Privacy Policy. Your downloading of the system image and use of the device software may also be subject to certain third-party terms of service, which can be found in Settings > About phone > Legal information, or as otherwise provided.
Acknowledge
I have read and agree with the above terms and conditions.
Was this helpful?

Google Play services
"lynx" for Pixel 7a


20:35
Install
Method
LET'S GO
Select and Patch a File
2024.2.3 Magisk v27.0
Full Changelog:




Today
Gallery
platform-tools-latest-windows
19/07/2024 12:19
File folder
magisk_patched-27000_AC3G8
19/07/2024 20:34
Disc Image File
8,192 KB
Desktop



Take a break and explore your fresh new OS, play around with settings, observe what's new, etc. :)

Step 3: Easily install all FOSS Applications and replace proprietary

FOSS - **"Free and open-source software is software that is available under a license that grants the right to use, modify, and distribute the software, modified or not, to everyone free of charge."**

We will use Fdroid - **"An open-source app store and software repository for Android, serving a similar function to the Google Play store."** Let's first start off by updating GrapheneOS stuff, then installing a third-party client Fdroid App.

1. Open "Apps" application - Install all and update any applications, Vandium will be your new Chrome >> Copy this link in the Vandium Browser https://f-droid.org/repo/com.looker.droidify_630.apk >> Download and install **"droidify"** (third-party client for Fdroid)
2. Open droidify >> set up and allow required permissions >> customise settings to install apps via root (if you rooted phone) >> Install the following apps (recommended):

Basic Apps:

- **Simple SMS Messenger**
- **Simple Calendar Pro**
- **Simple File Manager Pro**
- **Simple Dialer**
- **Simple Clock**
- **Simple Contacts Pro**
- **Simple Camera**
- **OpenCalc** (Calculator)

ALL the rest of FOSS, I have installed:

- **App Manager** - Android Package Manager (We will discuss this in Step 5)
- **AdAway** - System-wide ad, trackers and malicious stuff blocker on your phone - Very USEFUL! (Root only)
- **Rethink: DNS + Firewall** - Change DNS, Block Malware, Monitor Network Traffic - Very USEFUL!
- **CPU Info** - Device's hardware details: CPU, Memory, etc (We will use this in Step 4)
- **Xtra** - Alternative Twitch Client (No ads, cleaner, better UI)
- **FairEmail** - Alternative to Gmail (PGP Encryption, privacy focused)
- **Mull** - A forked, very hardened browser of Firefox
- **TimePlanner** - Track Tasks schedule plan of day.
- **Bitwarden** - Password Manager
- **Notally** - Very nice notes app
- **Neo Backup** - Backup applications and their data - VERY USEFUL! (root only, We will discuss this in Step 5)
- **Breezy Weather** - Weather App
- **Hacki** - Hacker and Tech news
- **Organic Maps** - Alternative to Google Maps
- **FlorisBoard** - Android Keyboard
- **AntennaPod** - Listen to Podcasts
- **RootlessJamesDSP** - Be the DJ of the music you're listening to 😎
- **KISS Launcher** - A much better android launcher and saves battery
- **Aurora Store** - A full replacement of the google play store with all apps
- **Infinity** - A reddit client
- **Aegis Authenticator** - MFA/2FA app to manage tokens for online services
- **Nextcloud** - A cloud solution client
- **Molly** - A messenger fork of Signal, focused on enhanced security
- **Find My Device (FMD)** - Locate and control your device remotely if lost
- **AnkiDroid** - Flashcard based study app
- **QuickNovel** - Read Novels and Books for free
- **Dantotsu** - Watch Anime and Read Manga for free
- **ViMusic** - Listen to all music for free in the background (Better than Spotify)
- **Tubular** - A lightweight YouTube Client app, forked off Newpipe X Sponsorblock (No ads, private)
- **Record You** - Screen Recorder
- **PDF Doc Scan** - Scan documents and save them as PDF
- **VLC** - Video Player
- **Yet Another Call blocker** - Block Unwanted Phone Calls
- **OnionShare** - Send files over the Tor network securely.

Note: If you can't find a few of these on the droidify store (Fdroid), then find them on github and download the apk file. □

Step 4: Easily Install Nethunter Lite (root) on Graphene

If you wish to install non-rooted or having troubles - You can refer to this David Bombal's rootless video and the steps should be the same for Pixel Phones > https://youtu.be/KxOGyuGq0Ts?feature=shared. If you are using another rooted device, you may have to disable some verification and encryption via TWRP https://xdaforums.com/t/deprecated-universal-dm-verity-forceencrypt-disk-quota-disabler-11-2-2020.3817389/

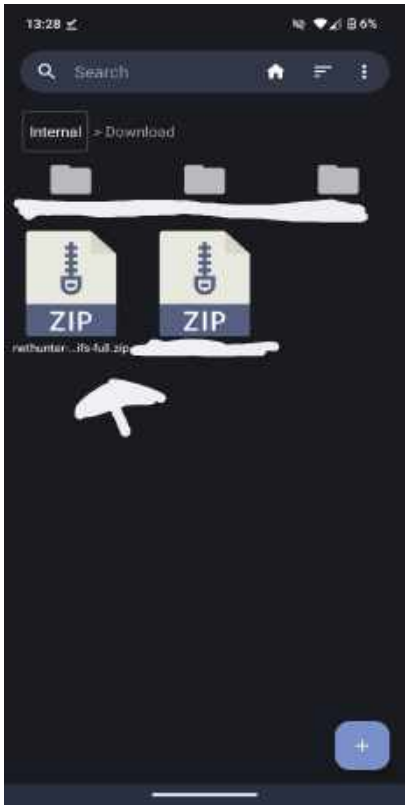
1. Using the FOSS CPU Info application >> Scroll down a bit > Look at ABI >> My one says **"arm64-v8a"**, so that is my phone's CPU architecture.

2. Now download your nethunter lite version zip from here (Full version and not the "torrent" or "sum") on your pixel https://www.kali.org/get-kali/#kali-mobile
3. Now open Magisk >> Modules >> Install from storage >> Select the downloaded zip file >> Magisk will handle the extraction and installation process from here (25 min)
4. Click reboot once magisk finishes >> 4 apps are installed >> Open them, grant permissions and superuser access >> Open nethunter >> Kali Chroot manager >> Start Kali Chroot
5. Now you're able to access all the nethunter tools :), make sure to stop kali chroot after you're finished.

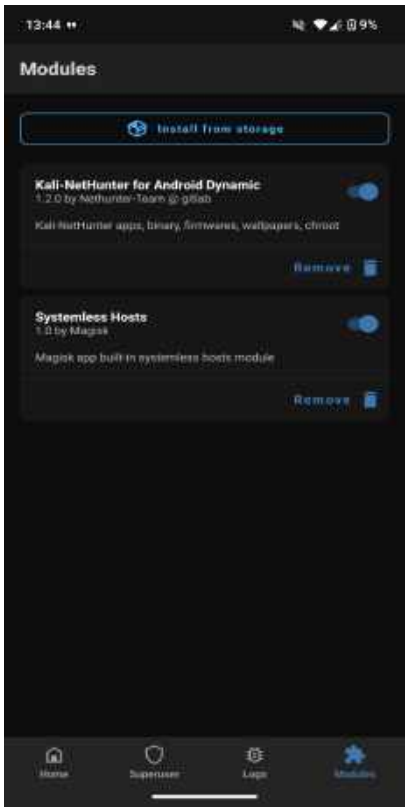
Few things to note: An Important thing to mention that Nethunter terminal doesn't have connectivity as I think GrapheneOS blocks traffic related to DNS, I believe for security. If you experience that issue, consider using Termux and/or a rootless installation.


13:30
Hardware
CPU
GPU
RAM
Storage
Core 5: 400MHz
0Mhz
2348Mhz
Core 6: 500MHz
0Mhz
2850Mhz
Core 7: 500MHz
0Mhz
2850Mhz
SoC name
Unknown
ABI
arm64-v8a
Cores
8
ARM Neon
Yes
L1d cache
32.00KB
32.00KB
32.00KB
32.00KB
32.00KB
32.00KB
32.00KB
32.00KB
L1i cache
32.00KB
32.00KB
32.00KB
32.00KB
Hardware
Applications
Temperature
Settings

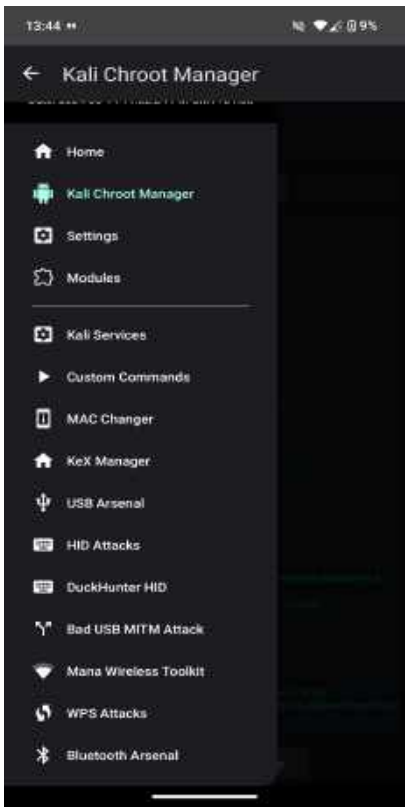

11:19
www.kali.org/get-kali/#k
Kali NetHunter Documentation ›
NetHunter Lite
Nexus
Xiaomi
Nokia


13:28
Search
Internal
> Download
ZIP
ZIP


13:42
Installation
Done!
Unpacking the installer...
Checking for previous version of NetHunter apps and files
Installing apps...
- Installing NetHunter.apk
- Installing NetHunterTerminal.apk
- Installing NetHunter-KeX.apk
- Installing NetHunter-Store.apk
Done installing apps
Installing NetHunter BusyBox...
Installing busybox_nh-1.25...
Installing busybox_nh-1.32...
Setting busybox_nh-1.32 as default
Creating symlinks for busybox applets
Installing NetHunter wallpaper
Found screen resolution: 1080x2400
NetHunter wallpaper applied successfully
Installing NetHunter boot animation
Copying nano highlights to /system/etc/nano
Copying terminfo files to /system/etc/terminfo
Copying Wifi firmwares to /system/etc/firmware
Installing /system/bin binaries
Installing hid-keyboard to /system/xbin
Installing init.d scripts
Installing /system/addon.d backup scripts
Symlinking Kali boot scripts
Granting required permissions to NetHunter app
Found Kali chroot to be installed: kalifs-arm64-full.tar
Extracting Kali rootfs, this may take up to 25 minutes...
Kali arm64-full chroot installed successfully!
Unmounting...
Done!
Kali NetHunter is now installed!
Please update the NetHunter app via the NetHunter Store to work around an Android 4
permission issue and run the NetHunter app to finish setting everything up
Reboot


13:44
Modules
Install from storage
Kali-NetHunter for Android Dynamic
1.2.0 by Nethunter-Team @ gitlab
Kali-NetHunter apps, binary, firmwares, wallpapers, chroot
Remove
Systemless Hosts
1.0 by Magisk
Magisk app built in systemless hosts module
Remove
Home
Superuser
Logs
Modules


13:44
Kali Chroot Manager
Change path
/data/local/nhsystem/
kali-arm64
Running
[13:44:03] [!] chroot path is set to -> /data/local/nhsystem/kali-arm64
[13:44:03] [!] verifying chroot environment...
[13:44:03] [+] seems you have kali chroot installed already!
[13:44:03] [+] proc was mounted already
[13:44:03] [+] sys was mounted already
[13:44:03] [+] dev was mounted already
[13:44:03] [+] devpts was mounted already
[13:44:03] [+] system was remounted already
[13:44:03] [+] sdcard was mounted already
[13:44:04] [+] Chroot environment has been set and started
[13:44:04] [!] You may click "STOP KALI CHROOT" to shutdown the chroot environment
Add Metapackge
Stop



Step 5: Additional Configurations - Permissions, Backups, Modules

Permissions (App Manager)

Android app permissions - Android app permissions control what data and resources an application can access. GrapheneOS enhances permission management with additional features in app settings. We'll dig into advanced permissions, typically hidden from standard settings, App Manager will help us to view detailed permission usage and app components, providing better control over privacy.

1. Open App Manager (You should've installed in Step 3) >> Grant all perms, setup, etc >> Click on an app you've installed.
2. Explore the **"Uses Permissions"** tab by sliding left >> Disable all permissions the app doesn't need. (RESEARCH FIRST!)
3. Lets view the **"Activities"**, **"Services"** and **"Receivers"** tabs. (As you can see, there is alot going on with just one app, depending how heavy it is. Disabling anything can lead to application crash.) >> Now disable all the ones that say **"TRACKER"** from those tabs >> Drag and slide the blue button (just tapping it won't work).

An easier method to disable all trackers for all applications - Go back >> Click the 3 dots >> 1-Click Ops >> Block/unblock trackers >> Select Yes >> Block

Bonus: A very cool thing you can do on App Manager is enable VirusTotal checks on APK files before it installs the application.

Go back >> 3 dots >> Settings >> Installer >> Select Installer App as **App Manager** >> Enable Block trackers (This will auto-block trackers for all new apps you install) >> Go back >> Set a VirusTotal API Key (You can do this by signing up to VirusTotal and grabbing a free API Key)

https://github.com/user-attachments/assets/ad4ee925-8fb7-435b-a0f8-1cceff5b5500

Note - If you want to learn more, you can use the User manual, clicking the 3 dots. I just showed a bit on what I know about it.

Backups (Neo Backup)

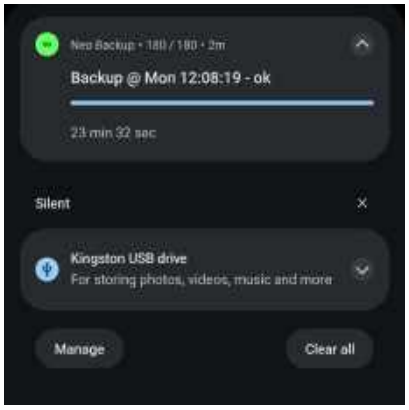
Backing up and restoring app data is probably one of the major reason, I'm going down the android rabbit hole I think. You can't backup app data without root, this is because android environment is sandboxed and applications are isolated. Additionally, I think, the directory to the data path is blocked, `/root/data/data`.

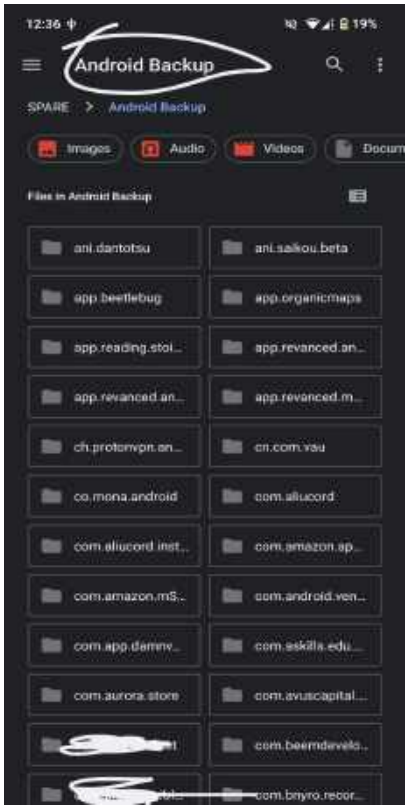
If you find an app, you're able to backup data without root access then it could be considered a vulnerability depeding on context, if it has a VDP/BBP, report responsibly. The impact is **OWASP-M2: Insecure Data Storage, Data tampering, Possible Information Disclosure and Bypassing Security Controls.** There are also techniques to manually check if an app allows backups through RE all the files which is more bugbounty focused.

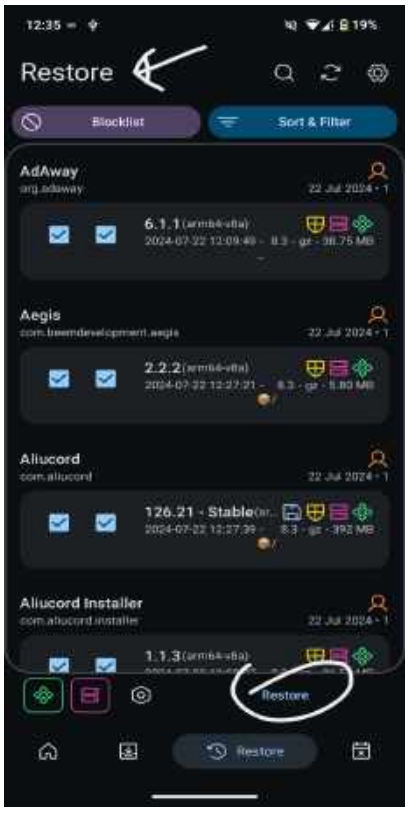
Requirements - Rooted, Minimum 32GB USB Stick, USB Type A to C adapter

1. Plug in your USB stick into your phone using an adapter >> Open the Neo Backup application (Installed in step 3) >> Let's go >> Choose Backup Folder >> In files >> 3 lines >> Choose your external drive >> Create a folder called backups and click into it >> Use this folder >> Enable notifications on Neo >> Optionally enable encryption.
2. Now go into **Backup** tab >> Click **Sort & Filter** >> Diselect **System Apps** >> Apply >> Click the **green (APK Files)** and **pink (Data)** outlined boxes >> Now on search >> filter out these applications by unticking both boxes: **Neo Backup**, **Magisk**, **Google Play Store**, **All for 4 nethunter applications** >> Click Backup
3. The process should take around 20-30 minutes, after completion >> Click the third tab restore and Neo backup will list all the apps that have a backup in that USB folder.
4. To restore the apps/data on a different/same android device >> Install Neo Backup >> Plug the same USB stick and choose same folder you backed up on >> Restore.

https://github.com/user-attachments/assets/1cf41cee-47b3-4b23-8d63-a2bbcfab4a16







Congrats if you made it this far, I tried to keep the steps very simple and professional for everyone, credits to alot of research and forums. I tried to make an All-In-One Solution for Individuals.

Modules

Now i ain't gonna lie, haven't done this one cuz I've not found it necessary. I only ever ramble about shit I learnt and showcase shit I done □. I'll update this if I play with the EdXposed

module.

Bonus: MDM

Back to being professional.. MDM is a software used by organisations to secure, manage and control mobile/tablet devices centrally. A good example is Microsoft Intune, which also supports desktops. Now for companies, it's important to implement policies and prevention measures on their MDM to prevent device rooting/jailbreaking.

Impact: A technical employee or staff (insider threat) can essentially make an MDM useless and redundant, simply by rooting/jailbreaking their work device, leading to physical theft with no possible way to locate as it'd be off MDM's radar. **Solution:** A good solution would be to completely disable developer options, USB debugging or system settings entirely on all mobiles/tablets by creating a profile. Additionally, setup alerts and monitoring. Although, I'm not too sure how this process would work for BYOD devices.