CleanYourTV

Remove the ads, bloatware and tracking from an Android TV or Google TV with one prompt, and install a clean Apple TV style home screen. Overview and step-by-step guide.

1. What it is

CleanYourTV is a free, open-source project by Gathin (GitHub: Gathin23). It connects to your TV over Wi-Fi using Android's debug bridge (ADB), disables the manufacturer's telemetry, ad services, content rows and preloaded apps, and installs a small native launcher as the new home screen. A coding assistant such as Claude Code does the work from a single prompt included in the repo, and checks each step with screenshots taken from the TV.

It started on 17 October 2026 as a before-and-after post of the creator's parents' TV. The repo was first called **tv-vibehack** and was renamed **cleanyourtv** on 19 October. Old links still work.

Item	Detail (as of 1 Nov 2026)
Repository	github.com/Gathin23/cleanyourtv (website: cleanyour.tv)
Popularity	48.2k stars, 6k forks, about 4,600 commits. #1 trending repo on GitHub for 19 to 25 Oct 2026.
Licence	MIT
Coverage	Tested debloat lists for 398 TV models from 104 brands. The first four days produced 170+ models from 200+ contributors.
Launcher	About 572 KB, plain Java, 31 languages, no network permission, no background services. Needs Android 8.0 (API 26) or newer on the TV.
Works with	Claude Code, Codex, Gemini CLI or Antigravity CLI. Can also be done by hand without any assistant.
Time needed	About 15 minutes with the repo cloned, according to the README.

2. How it works

Android TVs have the same debug bridge as Android phones. Once you switch on USB debugging in the TV's developer options and allow your computer once, the rest is standard Android tooling. No root, no cable and nothing is uninstalled.

Step	What happens
Back up	The list of installed packages and settings are saved first, in ~/.cleanyourtv/ on your computer, along with what Home pointed to.
Disable	Telemetry, ad services, content rows and preloaded junk are turned off with `pm disable-user`. Packages are disabled, not deleted, so every step can be undone.
Install	A small native launcher is built, installed and made the default home screen. The stock launcher is disabled only after the new one is confirmed working.

3. What you get

- A pitch-black home with a dock of your four favourite apps (4 to 7 slots are supported) and the rest of your apps in a grid that slides up when you press Down.
- A top shelf that names the focused app, with a glow taken from its artwork, and one floating glass highlight that moves from tile to tile.
- Frosted buttons for Inputs (HDMI ports, AV, Live TV), Wi-Fi (shows the network name) and Settings, plus a greeting, a thin clock and the date.
- No sponsored rows, no Discover prompts, no analytics from the manufacturer and no apps you did not choose.
- Optional extras: 15 themes you can push without a rebuild, a clock screensaver, an app library, and module cards for add-ons.

> **Limits to be clear about.** It disables system packages and replaces the home screen. It does not remove ads inside streaming apps. Android Private DNS pointed at an ad-blocking resolver (Android 9 and newer) blocks some in-app ad domains, but it does not remove YouTube ads.

4. Will it work on your TV?

Platform	Result
Android TV / Google TV	Full support: debloat and launcher. Includes TCL, Sony, Xiaomi/Redmi, Philips, Skyworth, Hisense, OnePlus, Vu, Panasonic, Nvidia Shield and many more.
Fire TV	Debloat only. Fire OS will not give Home to a third-party launcher. On the 2015 Fire TV Box even the debloat is refused.
A few China-market TCL sets	Refuse to install any launcher. Their lists say so in the header.
Samsung (Tizen), LG (webOS), Roku, Vizio, Hisense VIDAA	Not supported, because there is no ADB. The project links a page for turning off tracking from the remote: cleanyour.tv/not-android

Lists with the most models: TCL (80), Sony (44), Xiaomi (37), Philips (21), Skyworth (15), Hisense (14). Search your exact model at cleanyour.tv/tvs, or look in the examples/ folder of the repo. If your model is not listed, the prompt discovers the bloat on your TV itself.

5. Step-by-step guide

5.1 Before you start

- An Android TV or Google TV, switched on, on the **same Wi-Fi** as your computer.
- A computer (macOS, Linux or Windows) with a shell. The assistant builds the launcher there, so it needs Python 3, a JDK from 17 through 22 (a full JDK, JDK 23 and newer breaks the build), and Android SDK platform 34 with build-tools 34.
- A coding assistant account (for example Claude Code on an Anthropic plan). Without one, use the manual route in section 6.
- Your TV's exact model name, from Settings, About.
- Know what is plugged into the TV's USB ports. Some boxes cut power to USB while USB debugging is on, and a USB Ethernet adapter can drop too. Plan to finish in one sitting.

5.2 On the TV

- Open Settings, Device Preferences, About. Press OK on **Build** seven times to unlock developer options.
- Go back to Developer options and turn on **USB debugging**. If your TV shows a Network debugging or ADB debugging switch, turn that on too.
- Keep the TV on for the whole session.

5.3 On your computer

```
git clone https://github.com/Gathin23/cleanyourtv
cd cleanyourtv
claude
```

Cloning matters. The assistant reuses the repo's `tools/disable.sh` and `launcher/build-and-install.sh` instead of writing its own, and that is where the backup, the Home guard and the restore record live.

5.4 Paste the prompt

Open `PROMPT.md` in the repo, paste the whole text into the assistant, and fill in your TV model where the prompt says *My TV model is:*. The prompt opens with this request: debloat my Android TV and replace its launcher with a clean Apple TV style home screen, work on the real device step by step, and verify everything with screenshots pulled from the TV. If your TV is Google TV or you are not sure, say so.

What the prompt makes the assistant do, in order:

Phase	What it does
Connect	Finds the TV on your network and connects over ADB on port 5555. If macOS blocks the adb binary, it uses a Python ADB client instead. If the TV only offers Wireless debugging (common on Google TV), it asks you for the pairing code and port.
Back up	Saves the package list, disabled-package list, system properties and settings, and writes a restore script before changing anything.
Debloat	Sorts packages into telemetry, content and ad services, background services, preloaded apps and streaming apps. Disables them in small batches, one command at a time, with a screenshot between batches. It never touches core Android packages or what the TV needs to play HDMI and the tuner.
Ask you	Before touching streaming apps, the voice assistant, Chromecast, Miracast or the update service, it asks which ones you use and disables only what you leave unticked.
Launcher	Checks the TV accepts APKs and is on Android 8.0 or newer, builds the launcher, installs it, makes it Home, then disables the stock launcher. It asks how many dock slots you want and in which order.
Verify	Presses remote keys over ADB, takes screenshots and checks that Home, Input and app buttons work. It then gives you a restore script and a README.

5.5 Follow along

- When "Allow USB debugging?" appears on the TV, choose Allow. The assistant tells you when to expect it.
- Answer the checklist: which apps you use, dock size and order, and whether to keep the voice assistant, Chromecast and firmware updates.
- On Google TV, press each remote button (Netflix, YouTube, Input, Settings) once the stock launcher is off, and tell the assistant which ones do nothing so it can remap them.

- Wait about 15 seconds after the last change before the reboot, then reboot once. Some firmware turns packages back on at boot, and the assistant notes those in the list header.

5.6 Finish: close network ADB

Port 5555 is unencrypted. On some firmware any device on your network can get a shell through it, so close it when you are done.

- On the TV, turn off Wireless debugging, Network/ADB debugging and USB debugging, and revoke the USB debugging authorisations.
- Reboot the TV.
- On your computer run `python3 tools/adb-closed.py <tv ip>`. It must report the port as closed. A "Connection refused" is the only answer that counts.

The launcher keeps working with debugging off.

6. Doing it by hand (no coding assistant)

Everything the assistant does can be run yourself from the repo folder. Choose the folder in `examples/` that matches your TV model. The example below uses the Xiaomi Mi TV 4A list. Commands are for macOS and Linux; the repo has PowerShell versions for Windows.

```
python3 -m venv tools/venv && tools/venv/bin/pip install -r tools/requirements.txt
export TV_HOST=192.168.1.50                  # your TV's IP address

tools/venv/bin/python -I tools/tv.py --probe  # model, Android version, firmware
tools/disable.sh examples/xiaomi-mi-tv-4a/debloat.txt   # apply a debloat list
launcher/build-and-install.sh                # build the launcher and make it Home

# Phone only, no computer? Use the community app: cleanyour.tv/app
```

The scripts protect Home: they only disable a stock launcher they recognise and skip it in a list until another launcher answers the Home intent. If you have no assistant and no computer, a community-built Android phone app does the same cleanup (cleanyour.tv/app).

7. Undoing it

```
TV_HOST=192.168.1.50 tools/restore.sh examples/<your tv>/debloat.txt
# Google TV: add the stock launcher package as a second argument
TV_HOST=192.168.1.50 tools/restore.sh examples/<your tv>/debloat.txt \
    com.google.android.apps.tv.launcherx
```

- **Keep the list you applied.** Restore replays the record saved before the change. A package you had disabled yourself stays disabled, and Home returns to its original component.
- The new launcher stays installed. Remove it in the TV's Settings, Apps if you want it gone.
- No computer? The TV's own Settings, Apps page can re-enable packages one by one.
- **After a firmware update** the disabled packages come back and Home reverts to the stock launcher. Open ADB again and run `tools/venv/bin/python -I tools/check.py examples/<your tv>/debloat.txt`. Add `--fix` to put it back.

8. Risks and known problems

Nothing is flashed, rooted or deleted. The real risk is disabling something the TV depends on, which is why lists are applied in small batches with a restore script ready. If a TV stops booting far enough to reach Settings, the repo says a factory reset from the recovery menu is usually the way back. Changing a TV's software may affect its warranty or support.

Problem	What to know
Home turns black or opens the wrong app (Google TV)	The setup wizard `com.google.android.tungsten.setupwraith` can take Home once the stock launcher is off. Disable it too and list it so restore brings it back. Check first that your Wi-Fi network was not saved during setup: disabling setupwraith deletes such a network and the TV goes offline. Re-join the network in Settings first.
Stuck with no launcher	Run `tools/tv.py 'pm enable --user 0 com.google.android.apps.tv.launcherx'` to bring the stock Google TV launcher back.
macOS: "No route to host"	macOS Local Network privacy blocks the adb binary. `tools/tv.py` avoids the binary, so use that.
Boot loop after a list	Never disable core Android packages (permission controller, package installer, systemui, settings and similar). An original Mi TV Stick failed to boot this way. `tools/disable.sh` refuses such lists.
Factory or production menus	Do not open packages named factory, production or fac menu, even to look. On some Haier/CVTE boards this kills the Input key and CEC.
TV refuses the launcher	Some China-market sets reject every sideload or any package declaring Home. Do the debloat half only.
Fake sites	Official links are cleanyour.tv and the GitHub repo. cleanyourtv.com and clean-your-tv.net are not run by the project, and there is no app in any app store and no prebuilt APK to download.

9. Checking the reel script against the repo

A few statements in the script need updating or softening. Everything below was checked on 1 Nov 2026.

Script says	What the repo shows
Over 16,000 stars	Outdated. It was 48.2k stars on 1 Nov 2026, and #1 trending on GitHub for the week of 19 to 25 Oct.
Works with over 170 TV models	Outdated. 170+ was the count after four days. It is now 398 models from 104 brands.
Removed all 25 bloatware apps	The 25-app figure appears in a fork called tv-home (mihirgupta0900), for one 2 GB TCL Google TV, along with memory use falling from 192 MB to 83 MB. I did not find 25 in the original repo. The count depends on the TV model.
Removed bloatware, ads and tracking codes	Packages are **disabled**, not removed or uninstalled, and that is why one command can restore them. It stops manufacturer telemetry and ad services and removes sponsored rows from the home screen. It does not remove ads inside streaming apps.

Script says	What the repo shows
Paste one prompt with your TV model	Accurate, but you also have to enable developer options on the TV, accept the debugging prompt and answer a short checklist.
Looks just like Apple TV, or code your own	Accurate. The design is Apple TV 4K style, and themes and config let you change colours, wallpaper, dock and columns.
Restore with one command	Accurate: `tools/restore.sh`, from the record saved before any change.
Not in the script	It works only on Android TV and Google TV. Samsung, LG, Roku, Vizio and Hisense VIDAA sets are not supported. Worth saying in the video so viewers with those TVs do not try it.

Sources

GitHub: github.com/Gathin23/cleanyourtv (README and PROMPT.md). Fork referenced for the 25-app figure: github.com/mihirgupta0900/tv-home. Project site: cleanyour.tv.