Episode Book

EPISODE #6182

Running a Private APT Repo for Family Machines

PUBLISHED	21 October 2026 at 15:15 UTC
DURATION	22:39
LISTEN	myweirdprompts.com/episode/private-apt-repo-family-25197/
HOSTS	Corn, Herman, Desmond, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.11
BOOK GENERATED	22 October 2026 at 07:46 UTC

SYNOPSIS

A zip file in a shared Drive folder is a software release process with no mechanism for the second release. So how do you run your own APT repository — and keep it private — without turning into your family's unpaid update service? This episode walks through the modern client-side setup: the new deb822 .sources format, why Signed-By matters more than ever, and the Sequoia-based signature strictness in Debian 13 that quietly broke a lot of third-party repos this year. Then the privacy question gets split in two — hidden versus authenticated — with honest verdicts on Cloudflare Access (stock apt can't use it), a Worker-plus-R2 basic-auth pattern, and the Tailscale serve route that works right up until someone's client is switched off. Plus the setup-package trick that makes key rotation a normal update, and how to get silent third-party updates via Origins-Pattern.

TRANSCRIPT



Corn

A zip file in a shared Drive folder is a software release process with no mechanism for the second release. That's Daniel's problem. He builds small Linux tools, packages them as Debian packages for Ubuntu, and wants some of them on his family's machines without becoming their unpaid update service. So: can he run his own APT repository, what's the convenient way to keep it private, and what else is out there? Last time we did this for Android, with private F-Droid repos and the rest. Same itch, different operating system.



Herman

Yes, he can, and it's less work than people expect. A repository is a folder of static files. The package files, an index listing them, and a signed release file that vouches for the index. No server software at all. We covered that basic machinery back in April, the tools like reprepro and aptly and the idea of plain static hosting. What I want to spend the time on is what's changed on the client side, because a repo that worked in twenty twenty-three can fail quietly on a fresh install today.



Corn

Quietly is the worst way for a family machine to fail.



Herman

It is. Start with how the source gets declared. The modern way is the newer deb eight two two format, a file ending in dot sources in the sources.list.d folder, with proper fields instead of one cramped line. And the field that matters most is Signed-By. It points at a specific key file, or carries the key inline, and says: this repository is trusted only to be signed by this key.



Corn

As opposed to what we did for fifteen years.



Herman

As opposed to dumping the key into a global trust store, where every repository you had ever added was suddenly allowed to vouch for every other one. If you leave Signed-By out, apt falls back to that global store. Bad habit, and the old apt-key command that fed it is



Herman (continued)

on its way out. I believe it's gone entirely in apt three point one, which ships with Ubuntu twenty-six oh four, but I got that from a secondary source, so hold it loosely.



Corn

And where does the key live?



Herman

The man page draws a line. Keys that a package manages go in the usr share keyrings folder. Keys the operator drops in by hand go in a keyrings folder under etc apt. Either way, the apt user has to be able to read the file. Half of the "why is apt telling me the key is not available" threads are just permissions.



Corn

Fine. Key in a folder, source file pointing at it. What's the trap?



Herman

The trap is the key itself. Debian thirteen verifies signatures with a Sequoia-based tool called sqv, and sqv refuses signatures that rely on SHA one when they bind a key to its identity. A lot of third-party repos broke. People reported it for OpenResty, NodeSource, the NVIDIA CUDA repo, Sublime. One report put the cutoff at the start of February this year, but that's a single source, so I wouldn't quote the date. The fix is on the repo operator: generate a new key with modern self-signatures.



Corn

So an old key lying around from some earlier experiment...



Herman

Throw it away. Make a fresh one. Ed twenty-five five one nine, or RSA four thousand ninety-six, with SHA two fifty-six self-signatures. Whether Ubuntu twenty-six oh four applies the same strictness by default, I couldn't confirm. Debian thirteen is the one I



Herman (continued)

know rejects it. But if his family are on Ubuntu or Mint, a fresh key costs him ten minutes and removes the question.



Corn

Ten minutes to avoid a mystery. I'd take it.



Herman

Now, private. And I'd like to separate two meanings, because I think Daniel is mixing them in a perfectly sensible way. One meaning is hidden: nobody who doesn't know the address can find it. The other is authenticated: you need a credential to fetch anything.



Corn

Hidden is a long URL.



Herman

Hidden is a long URL, yes, and for a family repo of half a dozen small tools it might even be fine. Signing helps you here. Because every package is covered by a signature, somebody who stumbles on the repo can't tamper with what your family installs. What they could do is read it. So the risk you're managing is confidentiality, not integrity.



Corn

Nobody is going to break in to steal my uncle's... what are these tools, Daniel? A disk-usage viewer?



Herman

Whatever they are, the question is whether he minds a stranger downloading them. If yes, apt does support authentication, just narrowly. There's a file in the etc apt folder called auth dot conf, or a directory of dot conf files, in the old netrc style. Machine name, login, password. It has to be root-only, and if you give no protocol it matches HTTPS only, which is what you want. And the path matching is literal. An entry for slash deb does not match



Herman (continued)

slash debian.



Corn

Literal as in it will fail without telling you why.



Herman

Literal as in you will lose an evening. The alternative is putting the credentials inside the URL in the sources file, but that file is world-readable, so every user on that machine sees the password.



Corn

A family laptop. Every user on it is also a family member who already knows the Wi-Fi password.



Herman

Fair, the threat model is mild. But a token in the URL is also the pattern the hosted services use, and there it's per-person and revocable, which matters when your cousin sells her laptop.



Corn

Right. So what's the clever solution everybody reaches for first?



Herman

Cloudflare Access. Daniel is already a Cloudflare person, it's free at his scale, it handles logins, it's the tidy answer. And stock apt cannot use it. Apt can't send custom headers. Cloudflare Access wants a login flow in a browser, or a service token in a header. Cloudflare themselves built a custom apt transport for this, a program that wraps their cloudflared tool, and the post is from twenty nineteen. It's built from Go source, and on a desktop it pops a browser for login.



Corn

So to give my mother updates, I first ask my mother to compile Go.



Herman

Which is the whole reason I'd rule it out. What I think does work, though I haven't seen a write-up specific to apt, is a small Cloudflare Worker in front of an R two bucket that checks basic auth. The files sit in R two, the Worker says "who are you", and apt, which does speak basic auth, is happy. Cloudflare documents R two as a public apt repository. The private variant is the one I'd call plausible rather than proven.



Corn

Plausible rather than proven is how my leaf medicine is described too.



Herman

Corn, I'm going to leave that there. There's a third way that skips credentials altogether. Keep the repo on his own server, and serve it only inside a tailnet. Tailscale has a serve command that publishes a directory over HTTP or HTTPS inside the network, with a certificate it provisions for you. Static apt repo, no passwords, no public address, nothing to scan.



Corn

And everyone in the family joins Daniel's tailnet.



Herman

That's the catch. Each machine needs the Tailscale client, signed in or invited, and running when the update check fires. Daniel, I'd ask myself honestly whether your aunt is going to keep a VPN-shaped thing turned on. When it's off, the update just doesn't happen, and nobody knows.



Corn

A security system that works by the family forgetting it exists. I admire the minimalism.



Herman

I should also say: I haven't confirmed how apt behaves over plain HTTP on a tailnet versus the HTTPS certificate route. Use HTTPS and avoid finding out. If it's his own machines and two or three technical relatives, the tailnet is the lowest-effort option on the table. For anyone who can't be trusted to leave a client switched on, it's the wrong one.



Corn

Which is most of the people he's trying to help.



Herman

And now, a word from our personal finance guru, Desmond.

D

Desmond

Desmond's Money Tip. A sandwich bag costs about three pence. If you rinse it out and use it for four days instead of one, you save roughly nine pounds a year. Over a working lifetime, that is three hundred and fifty pounds, which is one modest funeral.

D

Desmond

I have a peg line by the sink with six bags on it in order of age, and I know which one is the oldest, and I do not let guests near it.

D

Desmond

That is your money tip. Money is mostly just watching things wear out and deciding not to let them.



Herman

I can see Corn eyeing the sink drawer, and no. Back to the repo: there's a pattern that fixes the thing most likely to go wrong, which is the one-time setup.



Corn

The key and the source file again. How does a family member get those without opening a terminal?



Herman

They install a package. A small one, named something like Daniel's archive keyring. It installs his public key into the usr share keyrings folder and drops the dot sources file next to it with Signed-By already pointing at that key. That is the split Debian itself recommends, package-managed keys in one place, hand-managed ones in another. The family double-clicks one file, once, and from then on the repo is just there.



Corn

Okay, that's smarter than I expected, because the key is now a package. Which means...



Herman

Which means when you rotate the key, and you will, probably the first time you realise the old one was a mistake, it ships as an update. Nobody edits a file. Google Chrome does something like this, though that's from my memory and I didn't re-check it. The setup package is also where I'd put the second half of the problem.



Corn

Which is that installing the repo isn't the same as getting the updates.



Herman

Right. Ubuntu desktops already install security updates unattended, but only from Ubuntu's own origins. A third-party repo isn't on the list, so Daniel's family would get his update notice in the update manager and click it or ignore it. For silent updates there's a



Herman (continued)

setting called Origins-Pattern for the unattended-upgrades tool. It matches on origin, label, archive, or site, and the site match accepts globs. If he doesn't set a suite or a label in the release file, site is basically the only handle.



Corn

He ships that in the same setup package.



Herman

Yes, as a small drop-in file in the apt config directory. It adds his pattern to the existing list. And he tests it on his own machine with the dry-run mode of the unattended-upgrade command, which tells you what it would have installed. You find out the value to match from apt-cache policy.



Corn

Dry run. You're telling me to test something before sending it to my family. Radical.



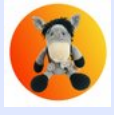
Herman

Two small tells here, Corn. You said "my family" like it was a production environment, and it is.



Corn

Most of my production environment doesn't read the changelog either.



Herman

Now, building it. The nice part is that Daniel doesn't have to do the signing on his laptop. A GitHub Actions workflow can build the package, update the index, sign it, and publish. The tools: aptly can publish straight to an S three endpoint but needs the signing key available. There's a Ruby gem called deb S three that updates the index files directly on S three, no local copy of the repo between runs, which suits a CI job. And one real-world example I came across used plain dpkg-scanpackages and apt-ftparchive, a signing key



Herman (continued)

with no passphrase stored as a repository secret, publishing to GitHub Pages.



Corn

Pages. That's the free host everyone sees first.



Herman

And the one I'd steer him away from, because Pages is public. The site is a public website, even when the repo behind it isn't. I believe a private-only Pages site exists on the Enterprise plan, but I haven't verified that. And GitHub Packages, the thing with the promising name, has registries for npm, Ruby gems, Maven, Docker images and so on, but not Debian. I didn't find an explicit statement that apt isn't supported. It's just absent from the list.



Corn

So the free, easy one is public and the one with the right name doesn't do the job.



Herman

That's the whole page, yes. R two or a plain object store is where the CI job should push. The signing key in the secret is a trade-off, because anyone who can edit the workflow can sign things, but for family tools it's a reasonable one.



Corn

What if he doesn't want to run any of it? Hosted options. Who does this for money?



Herman

A few, with a caveat that I'm unsure about several of the prices. Gemfury, now called Fury, I did check: the free tier has no private packages. The personal plan is nine dollars a month for one collaborator, and it lists a private Debian repository. Consumers fetch with a token. Cloudsmith does private Debian repos with entitlement tokens, read-only and revocable per person, embedded in the URL. That's attractive for family because you can



Herman (continued)

cut off one cousin. Their pricing, though, I saw conflicting numbers, so I'd leave the paid tier out of it. Their free tier looked small, a few hundred megabytes of storage, and the free open-source plan is only for public repositories.



Corn

A few hundred megabytes. How big are Daniel's tools?



Herman

Probably small. Which makes me think a free tier could hold them, but I wouldn't promise that without checking what the free tier allows for private repos, and I couldn't confirm it. Packagecloud exists too, and I couldn't find solid pricing for it.



Corn

Nine dollars a month, for... let me do the arithmetic. A hundred and eight dollars a year to avoid renting a five-dollar server.



Herman

Which is the actual comparison. A small VPS or an object bucket costs a few dollars a month or less, but you operate it. Fury costs a bit more and somebody else keeps the lights on. Daniel's call, depending on how much he wants to be the person who fixes it at midnight.



Corn

I'd pick the one where I'm not the person.



Herman

You pick the one where you're not the person for everything. Corn, does that include your own Mongolia mirror?



Corn

It's hosted in Mongolia. Obviously the uptime is excellent.



Herman

There are two more official routes for completeness. Launchpad has private PPAs, but you ask Canonical's commercial team and they grant it case by case. A twenty twelve post quoted around two hundred and fifty dollars a year, and that's very likely out of date. And Snap has visibility settings. Unlisted means anyone who knows the name can install it, but it doesn't show up in search. Private means the owner and collaborators by email address. The paid private store tier is aimed at device makers and negotiated with Canonical.



Corn

Which is why nobody's family is running that.



Herman

Then Flatpak, which is the serious alternative, and I think the only one with a real case against the deb for part of his list. If a tool is a desktop app with a window, Flatpak gives you updates inside GNOME Software, on Ubuntu, Fedora, Mint, anything. Hosting is static too: you run a build-update-repo command, optionally generate static deltas, and publish a dot flatpakrepo file that carries the key and the URL.



Corn

So what stops me recommending it for everything?



Herman

Privacy. Flatpak has no built-in basic auth. You can put credentials in the URL, but they show up in the output of the remotes command. The proper route is writing a custom authenticator, or bearer tokens, and the references I found are old forum threads. So it's the best update experience with the most awkward lock. For command-line tools, which is what Daniel mostly describes, the deb is simply the right format anyway. There's also AppImage, which is easy to hand someone and has no update story unless you add one.



Corn

AppImage is the zip file in the Drive folder with a nicer icon.



Hilbert

Does a family member ever get to refuse one? Reuben Tallis, fourteen Cutler Road, says refusal is a right. Herman, you keep saying "dot sources" like it's a town you've been to. Reuben has a pony called Brendan. I do not own Brendan.



Herman

On what Daniel's describing, nobody refuses. The update lands overnight and they find out afterwards.



Hilbert

Brendan carries my floppy disks to my cousins on Tuesdays, in a saddlebag Reuben stitched himself. Four towns, seventy-two disks a round. The arrangement is forty eggs a disk. Reuben keeps hens. I'm the one who pays, not my cousins, because Reuben does not recognise my cousins. He recognises Brendan and he recognises me. Brendan is also the registered recipient of the software at the Dunmurry post office, under his own name. The post office agreed to it.



Corn

Forty eggs for one point four four megabytes. That's a lot of omelette per megabyte.



Hilbert

Two thousand eight hundred and eighty eggs a round. They're in my mother's bath, the upstairs one. The latest is that Reuben has stopped counting disks and started counting files. Eleven on a disk. Fifty eggs a file, so five hundred and fifty a disk, by his sums, and he won't show me the sums.



Herman

Per-file billing means the way to pay less is fewer files. Pack all eleven into one archive and it's one file, fifty eggs.



Hilbert

Reuben says an archive is a file, and Brendan won't carry anything he can't hear rattle. The bridge flooded in August. Brendan wouldn't cross, so the August version reached Dunmurry in the first week of October. Five weeks. She ran a different version from the other three the whole of September and Reuben said that was her choice. I wrap each disk in a tea towel. Claws.



Corn

Send only what changed. Most months that's one file, so fifty eggs instead of five hundred and fifty.



Hilbert

One file, yes. Reuben counts a file as changed if I've looked at it, so I'd stop looking.



Herman

Then the recommendation, with a clear default. Build the package in GitHub Actions, sign it with a fresh modern key, and publish the static files to an object bucket, R two or whatever he already uses. Put a small basic-auth Worker in front if he wants real authentication. Ship one setup package carrying the key, the sources file, and the unattended-upgrades pattern. Each family member installs it once and never thinks about it again.



Corn

Authentication is where I'd hesitate. Herman, would you rather my mother typed a password into a root-only file, or never met one at all?



Herman

Never met one, for a non-technical family. If Daniel decides hidden is private enough, a long unguessable address inside the setup package removes the credential step entirely, and because everything is signed, the worst a stranger can do is read the packages. For anyone technical enough to cope with an auth file, the basic-auth Worker is the better lock. Which would you choose, Corn?



Corn

The long address. My mother has enough passwords, and I have enough phone calls.



Herman

Then the alternatives, when the default isn't right. Zero operation: Fury at nine dollars a month, a token per person. Every relative technical and already on Tailscale: the tailnet, no credentials at all. A desktop app with a window, on mixed distributions: Flatpak, accepting that the private part is awkward. Everything else is a variation on those.



Corn

Default or clever?



Herman

Whichever still works after you've forgotten how it works. That's the boring bucket.



Corn

The cutting-room floor. Launchpad private PPAs: you email Canonical's commercial team and they decide per project. There's no sign-up button.



Herman

The oldest official route to a private Ubuntu repo sits behind a person with an inbox.



Corn

Daniel, aim for the last machine in the family being one overnight cycle behind, not five weeks. Thanks as always to our producer, Hilbert Flumingtop.



Herman

This has been My Weird Prompts. You'll find every episode at my weird prompts dot com. See you tomorrow.