Episode Book

EPISODE #6578

How Air-Gapped Linux Systems Actually Get Updated

PUBLISHED	13 November 2026 at 23:57 UTC
DURATION	27:15
LISTEN	myweirdprompts.com/episode/air-gapped-linux-updates-53425/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	25 cited by the research (listed at the end)
BOOK GENERATED	14 November 2026 at 06:47 UTC

SYNOPSIS

Almost nobody runs a special air-gapped operating system; instead, enterprises deploy mainstream Linux like Ubuntu with Landscape or Red Hat with Satellite in disconnected mode. The real engineering challenge is moving updates across the gap without breaking it. Because Landscape cannot generate diffs, keeping offline Ubuntu machines patched means physically carrying hundreds of gigabytes of repository data every time. From Red Hat's hand-carried subscription manifests to purpose-built distros like Offline Tools OS and COLDIRON OS, this is how disconnected machines stay alive.

TRANSCRIPT



Corn

How much data does it take to keep a computer with no network updated? And the answer, for Ubuntu, is hundreds of gigabytes. Every time.



Herman

Every single time.



Corn

That's the hook. Daniel's prompt is part two of our locked-down Linux run — last time was the general lockdown story, and now he's asking about the machines that never touch a network at all. Signing keys, sensitive documents, a newsroom's secure viewing station, industrial and government kit. What are they actually running? Is it just Ubuntu or Red Hat with the networking torn out, or does somebody build a distro for this on purpose? And then the part I find interesting: how does anything get in or out? Software, updates, data — without the gap quietly stopping being a gap.



Herman

And the honest answer starts with a disappointment and then gets better. Because the first thing you find is that almost nobody runs a special air-gap distro. The enterprise pattern, everywhere, is a mainstream distro — Ubuntu, RHEL, Debian — with the management infrastructure around it deployed in disconnected mode. Ubuntu with Landscape, Red Hat with Satellite, or Debian with an internal mirror. The gap is procedure, not product.



Corn

Which will surprise people who assume the words "air-gapped" on a procurement form imply some hardened special-purpose thing.



Herman

Strictly, it's physical isolation — no network interface connected to any other network. No Ethernet, no Wi-Fi. But even Canonical's own documentation splits it into two flavours: a "restricted network airgap", where the inside environment has controlled one-way access to external endpoints, and a "fully disconnected airgap", where there's no connectivity at



Herman (continued)

all and everything moves by hand.



Corn

So it's a spectrum with a reputation for being a switch.



Herman

That's most of the industry's security vocabulary in one line. And here's the thesis I'd plant early: the interesting engineering in this whole space is not the distro. The distro is the easy part. The interesting part is the transfer process — the mirrors, the manifests, the signed metadata, the removable media, and increasingly actual hardware. Everything after this is me proving that.



Corn

Then start with the boring proof. What does Canonical actually tell you to do?



Herman

Landscape, their management server, supports three topologies for an air-gapped deployment. One: a single Landscape Server sitting inside the air gap, if your firewall rules allow it to reach some external endpoints. Two: the server outside, clients inside. Three — and Canonical's own wording here is "provides maximum network isolation" — two servers. One online outside the gap, one offline inside it. The outside one talks to the world, and you carry the repository data across to the inside one by hand.



Corn

A server inside the air gap that has firewall rules allowing it to reach external endpoints. It's a gap with a door, a welcome mat, and a nameplate.



Herman

Canonical is at least honest that it's the less painful option. And Red Hat's equivalent is Satellite, and the mechanism there is my favourite piece of bureaucracy in this whole story. A disconnected Satellite doesn't authenticate against anything. Instead you create, on their customer portal, what's called a subscription allocation, and from that you download a subscription manifest — their own words, "a set of encrypted files containing information about your subscriptions." You then physically carry that manifest to the disconnected Satellite Server and upload it. There's a proper command for it — hammer subscription upload — and that's how the machine proves it's allowed to exist.



Corn

A licence file carried by hand. Sneakernet as a licensing model.



Herman

It's a receipt that walks. And Satellite itself can be installed from an ISO you downloaded elsewhere, though Red Hat is careful to say that's not recommended in any other situation, because an ISO image might not contain the latest updates and fixes. Which is the whole tension of the disconnected world showing up in one caveat.



Corn

That caveat is basically the entire second half of this episode compressed into a support note.



Herman

It really is. So that's what organisations run — Ubuntu, RHEL, Debian, hardened, with the management plane doing the air-gap work. Now, the purpose-built distros do exist. They're real. They're just small.



Corn

How small?



Herman

Vermont Secure Computing makes one called Offline Tools OS. Debian-based, live system, and the thing that distinguishes it is where it removes the network — not in the firewall, not in a config file, but in the kernel. The physical network stack is disabled at the kernel level. No Ethernet, no Wi-Fi, no USB tethering. They blacklist every physical network driver, and they do it twice — once with the normal blacklist directive, and once with an install directive pointing at a program that just returns false, so the driver can't be force-loaded either. If you run the command to list your network interfaces on that machine, you get exactly one: loopback. The one that talks to itself.



Corn

The most honest network interface. Loopback never lies to you.



Herman

It's the interface equivalent of a sloth. And the build ships with a reasonable offline toolkit — Xfce, Firefox, LibreOffice, GnuPG, an offline Wikipedia reader, compilers, even input methods for Chinese, Japanese and Korean. Somebody thought about actual offline humans using this.



Corn

But.



Herman

But. The marketing on that page undercuts itself beautifully. It calls itself, quote, "the world's most secure operating system," and then, a few lines down: "Unless of course your hardard uses drivers not in the list below, in which case all bets are off." Followed by "USE AT YOUR OWN RISK." In all caps.



Corn

The world's most secure operating system, unless you boot it, in which case all bets are off.



Herman

And the default credentials are user and live. Which is the sort of detail that tells you exactly who this is for — somebody who wants the machine to boot and does not care who knows how.



Corn

A padlock shipped with the key taped to it, for people in a hurry.



Herman

The other one worth naming is COLDIRON OS. This one's aimed at Bitcoin cold storage, so it's a single-purpose appliance rather than a general distro. Hardened Debian thirteen, built as a live image that runs entirely from RAM — there's a boot flag for that, and the point is nothing is ever written back to the boot USB, so powering off removes all state. Pull the stick out and the session never happened.



Corn

Statelessness as a lifestyle. Finally an operating system designed for me.



Herman

And the network story is even more absolute than Offline Tools. The kernel is compiled from source with no network device drivers at all and no loadable modules. Not blacklisted — absent. There is nothing to force-load, because the code isn't in the binary.



Corn

So that's the real fault line, isn't it. Offline Tools blacklists drivers; COLDIRON doesn't compile them. One is a promise the kernel makes to you, the other is a fact about the kernel. If the driver exists and is merely disabled, your air gap is a configuration setting — and configuration settings have an awkward history of being settings.



Herman

That's the case the purpose-built distros are making, and it's a good case. I'd take the other side for a second, though, because I think it's close. A hardened Ubuntu behind that disconnected Landscape gets security patches. COLDIRON is at version zero point three, "first product release," and the release artifacts are signed with what they call a one-shot key — generated for the signing, revoked immediately afterwards. Now, that's actually a defensible choice: the attack surface is that much smaller, and it's auditable.



Corn

A signing key that has already retired says something about the support model, though. It signed the release, closed the door, and left the building.



Herman

It does tell you nobody's standing behind it long-term. So you're trading a kernel that physically cannot network for an ecosystem that can't patch. That's not a free trade either direction.



Corn

And the NixOS people?



Herman

The NixOS angle is the third flavour, and it's clever. Because Nix builds the entire system closure ahead of time, you can bake the finished system into the install image itself. One project has a build target — the offline one — that puts each host's pre-built system and its disk layout script into the live store, so an air-gapped machine installs without ever resolving a dependency. Another one patches the graphical installer to stop it checking for an internet connection. And every one of these projects carries the same warning, and I want to read it because it's the whole niche in one line: "WARNING: this is experimental. DO NOT USE FOR ANYTHING OF VALUE."



Corn

The only honesty in this entire space, and it's in capital letters on a project README.



Herman

It's the correct warning! And then the negative finding, which I'd state as a finding rather than a gap: none of these purpose-built options claims any formal security certification. No Common Criteria, no FIPS, none of it. Which is why the enterprise answer stays exactly where it was — mainstream distro plus disconnected management, plus process, plus more process.



Corn

Which means we've now answered Daniel's first question and we can move to the part that actually matters. The distro is the easy part. The hard part is that nothing gets in or out — and that's where the real engineering lives.



Herman

This is my favourite part of the whole topic. Every source on this — Canonical, Red Hat, the independent guides — converges on the same architecture. You have a mirror outside, a controlled promotion boundary where somebody decides what's allowed through, and an internal repository inside. Removable media is the physical link. But the detail I love is the one guide that counts what has to cross the gap. Four things. And only one of them is packages.



Corn

I was going to guess packages and patience.



Herman

The packages, yes. Then the repository metadata — the index with the sizes and hashes. Then the signatures on that metadata, and on the packages themselves. And fourth, the key that verifies those signatures — which, and this is the line, "must arrive by a channel you trust."



Corn

Say that last part slowly.



Herman

If the verification key rides the same USB stick as the packages it's supposed to verify, the signature check is theatre. Whoever controls the stick controls what the check proves. You've built a very expensive ritual for trusting the thing you were already holding.



Corn

That's the supply-chain attack surface in one sentence. The gap isn't the USB stick, it's whether the key and the cargo ever shared a pocket.



Herman

And Corn — that's the good case. Then the degradation problem starts. Canonical is unusually blunt about this one. Landscape can't generate a diff between the repository you already have inside and the latest one outside. You cannot bring only the differences. You must bring the entire repository contents, every time you update. And the size of the Ubuntu repositories is, their words, hundreds of gigabytes of data per version of Ubuntu. They call the manual transfer "difficult and time-consuming."



Corn

Hundreds of gigabytes. Carried by hand. Every update. Somebody at Canonical wrote "time-consuming" knowing full well what they were describing.



Herman

And the full-disconnection setup has extra requirements on top. Identical repository structures on both sides of the gap, the same GPG key on both. You can't use the web portal to manage the inside server, because the inside server has no network — you drive it through the API. And the data path is a specific directory that you copy to, their suggestion, a DVD or USB drive, and back into the same path on the other side.



Corn

"Acceptable form of physical media for your environment" is the phrase of a document that has given up predicting what your environment is.



Herman

There's an escape hatch, though, and it's the most modern thing in all of this: stop moving packages entirely. Red Hat published an image-mode walkthrough last year — this is their bootable-container model, where the whole operating system is one image. You prepare a USB on an online machine, copy the container onto it with skopeo, preserving digests. Then you walk to the offline machine, copy it into local storage, and switch the system over to boot from it. Their stated benefit is that the machine you're updating is fully offline and air-gapped the whole time.



Corn

No dependency resolution on the far side. You're not installing software, you're replacing the animal.



Herman

The entire dependency graph is resolved before anything crosses the gap, which removes a whole category of "what does this package actually pull in" anxiety. The costs they admit: it's time-consuming across many devices, and it requires someone on site capable of deploying the update. A human, physically present, per machine, forever.



Corn

The demo numbers are good colour, too, aren't they.



Herman

A one-point-four-eight gigabyte update container, on a twenty-eight-point-nine gigabyte USB stick. RHEL ten and Fedora Workstation forty-two. So a real update fits on a pocket-sized stick with room to spare — the sneakernet version is manageable. It's the fleet that kills you.



Corn

That's the same conclusion the immutable distros reached from the other direction. If your upgrade is a whole image, it's inherently easier to carry across a gap than a graph.



Herman

RouterKit OS says exactly that — image-based installs make air-gapped setups workable. So that's software going in. Data going the other way is where the hardware appears. The baseline is removable media — SecureDrop's own term is a Transfer Device, and it's literally a USB drive or a DVD that you carry between machines. But above that, there's a whole formal category. Data diodes — hardware devices that use physical means, like optical isolators, to create separation that can't be crossed electronically. One direction. Always. Not by policy — by physics.



Corn

A one-way valve with a procurement code.



Herman

The government category around them is the Cross Domain Solution — Australia's cyber security centre quotes their security manual defining it as a system implementing comprehensive data flow security policies with high trust between different security domains, and they note that air-gapped networks only accept data via removable media or a secure one of those. And the NATO-grade diodes are fast. The vs-diode is approved up to German SECRET, EU SECRET and NATO SECRET, and moves data from the unclassified to the classified side at over eight gigabits per second.



Corn

Eight gigabits a second through a one-way valve. The gap between "walk it over on a stick" and that is the entire budget of this conversation.



Herman

SyBard's version three does one gigabit over fibre with cryptography on top of the unidirectional guarantee.



Corn

So if you have the money, the gap becomes a machine with a bouncer.



Herman

And now the turn, because the reason this whole topic got interesting is what happened to the most famous air-gapped Linux deployment in the world. SecureDrop's Secure Viewing Station — the machine a newsroom uses to decrypt source submissions. Its own documentation: never connected to the internet or a local network, only ever runs from a dedicated Tails drive. That's the machine people picture when they hear "air-gapped Linux." And the hardening guidance for it is the best reading in this entire subject. Physically remove internal storage and networking hardware — pull the wireless card. If the machine has network ports you can't remove, cover them with labels. Or, their words, "fill a port with epoxy to physically disable it."



Corn

Epoxy. The documentation has escalated from sysadmin to builder's merchant.



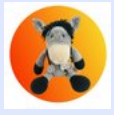
Herman

And my favourite line in the entire research — I want you to appreciate it. "Remove the speakers from the device, or just cut the audio cables if that's easier. This is to prevent exfiltration of data from the airgap via ultrasonic audio, which cannot be heard by humans."



Corn

Cut the speaker cables because the speakers might whisper. Even the machine's own voice box is a suspect.



Herman

It's not paranoia, either — we'll get to why that's a real threat. Even the clock is affected: there's no NTP on a machine with no network, so the admin sets the system time manually before creating the submission key, because a timestamp nobody can verify is a weaker timestamp.



Corn

Then the newsroom abandoned it. Tell that part properly, because it's the emotional centre of this whole episode.



Herman

SecureDrop Workstation development began in twenty-seventeen, and it replaces the system of Tails USB drives and air-gapped laptops with a single workstation based on Qubes OS — virtual machines on a networked laptop, with the decryption work in a network-isolated VM and untrusted files opened in disposable ones. As of late this year the project says it has transitioned completely to Qubes-based workstations, and the Tails version stops being supported by the end of twenty twenty-seven.



Corn

And not because it was broken.



Herman

Their own README is remarkably blunt. "Journalists must access a separate, airgapped device to even validate that a submission is relevant. The airgapped workflow is complex and slow, and may reduce the reliance on SecureDrop overall." In other words — the friction was making the security useless, because people were quietly using something else instead.



Corn

The gap didn't fail. The walk to the other laptop did.



Herman

They even document the operational failure: journalists accidentally booting the wrong device off the Secure Viewing Station's own USB drive — the boot stick for the most sensitive machine in the newsroom, used to start a regular one. And they're honest about the new risk. The Qubes approach stands and falls with the security of Qubes OS itself, which in turn may be impacted by Spectre and Meltdown-class CPU issues — VM isolation sitting on silicon that has had some very public questions asked of it. Hardware requirements are steep, too. Thirty-two gigabytes of RAM minimum, one hundred and twenty-eight gigabyte SSD.



Corn

Thirty-two gigs of RAM to replace a laptop with no network. That's the exchange rate between physics and software.



Herman

And I keep wanting to defend the old model, because I think the old model was right for its threat.



Corn

You defended it before I finished reading the docs, yes.



Herman

I did, and here's why. The covert-channel literature. The air gap sceptics have real material. AirHopper, twenty fourteen — malware on the air-gapped machine exfiltrating data over FM to a mobile phone sitting next to it. BitWhisper, twenty fifteen — bidirectional signalling through heat between two adjacent computers, no extra hardware required. GSMem, also twenty fifteen — cellular frequencies generated by a standard internal bus. Acoustic channels that work near or beyond the limit of human hearing, at ranges up to about sixty-five feet. ESET documented an espionage framework called Ramsay in twenty twenty that specifically collects documents from air-gapped networks. And Stuxnet was installed at Natanz in two thousand nine.



Corn

So SecureDrop's speaker-cable scissors were aimed at a real threat, and the air-gapped machine was never the fortress the diagram implied — it was a machine that leaks through its own physics unless you also control the physics.



Herman

Which is exactly why trading physical isolation for VM isolation is a genuine trade and not a free upgrade. You're swapping the covert-channel attack surface for the hypervisor attack surface. There's no option on the menu that's just "safe."



Hilbert

Hilbert's Mid-Show Haiku. The gap. Nothing crosses. A manifest sleeps in a coat pocket. The mirror waits, patient. The machine does not know it is alone.



Corn

Neither does the USB stick.



Herman

Speaking of USB sticks — the physical ritual is the part of all this that I think we've underweighted. We've been talking topologies and kernels, but every one of these architectures terminates in a person holding media.



Corn

You'd know. You're the one who carries the backup drive to the bank in your mouth.



Herman

It's a carrying case. It has a handle. And honestly, the SecureDrop README line about journalists booting the wrong device is the whole problem in miniature — the procedure assumed a person would remember which stick does what, at eleven at night, on deadline.



Hilbert

"Removable media." Herman, you've said that phrase six times this episode and it's the wrong phrase, and I'll tell you why. Media is what's on the stick. The stick is a stick. I ran transfers for a place that had fourteen of them in circulation, and the only thing that ever held up was the colour code. Blue carried nothing out. Green carried nothing in. Orange was the only colour allowed to touch the signing machine, and that rule survived three audits and one flood. The procedures all failed eventually. A tired man at eleven at night cannot get a colour wrong. He can get a checksum wrong. He cannot get blue wrong.



Herman

Fourteen sticks in circulation — that's a real operation. Did the colour code ever break down?



Hilbert

Once. I broke it myself, in March. Somebody had used the orange stick on a printer laptop and I needed a clean one at short notice, so I took a green one in where orange was allowed, and I logged it. I keep the log on paper, obviously. Every transfer I have ever done, date, colour, what was on it. I can tell you what was on the green stick on the fourteenth of that month. I have never once needed to, and I am not stopping. You two talk like the kernel is the hard part. I've watched a man with no kernel problems at all pick up the wrong stick because they were the same colour.



Corn

Two sticks the same colour. That's not a procedure failure, that's a stationery problem.



Hilbert

It's a management problem, Corn. The gap doesn't fail, the people do, and they were always going to. Build for that or don't build at all. Grunwald, who ran the desk next to mine, took eleven minutes to do a transfer that took me four, and I timed him, and I was right to.



Herman

The timing is the part I believe most, somehow.



Corn

It's also the part of the SecureDrop story nobody wrote down. Their README documents journalists booting the wrong device — but it never documents how the right procedure was supposed to survive year ten of a newsroom.



Herman

And the migration deadline makes it concrete. By the end of twenty twenty-seven, Tails-based SecureDrop is gone, and every newsroom that ran it is running Qubes VMs on a networked laptop instead.



Corn

So let me leave the thing I actually want to leave. If the most security-motivated air gap in the world — a newsroom protecting sources whose lives depend on it — traded physical isolation for VM isolation because the walk across the room was too slow, what does that say about every air gap that doesn't have a newsroom's threat model? The industrial ones, the government ones, where the procedure has to survive decades of staff turnover and nobody's decrypting anything more urgent than a Tuesday? Those gaps are still standing. And standing is all they're doing — no purpose-built distro out there claims any formal certification, and the enterprise answer is still a mainstream distro plus a lot of process. The gap was never a distro. It's a supply chain and a procedure wearing a computer as a hat.



Herman

And the procedure is the bit that has to outlive the enthusiasm of whoever set it up. That's true of air gaps and of most things.



Corn

If you've enjoyed this one, a review wherever you listen helps other people find the show — and our full archive, all six thousand four hundred plus episodes, is at my weird prompts dot com. Producer credit, as always, to Hilbert Flumingtop. If you want more of this, try episode two, Local STT For AMD GPU Owners; episode five, Fine-Tuning ASR For Maximal Usability; and episode twenty, Architectural AI. This has been My Weird Prompts — no question too obscure, no rabbit hole too deep. We'll be back soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Canonical/Ubuntu Landscape docs, How to manage repositories in an airgapped or offline environment, retrieved 2026-11-13 (notes apply to Landscape 25.10 and earlier; 26.04 LTS changes apply). [primary]**

   https://ubuntu.com/landscape/docs/how-to-guides/repository-mirrors/manage-repositories-in-an-air-gapped-or-offline-environment/

2. **Red Hat Developers, Armando Ortiz, How to deploy an image mode update in offline and air-gapped environments, published 2025-08-13, retrieved 2026-11-13. [primary]**

   https://developers.redhat.com/articles/2025/08/13/deploy-image-mode-update-offline-and-air-gapped-environments

3. **Red Hat Subscription Central 1-latest, Manage subscription allocations for disconnected systems, retrieved 2026-11-13. [primary]**

   https://docs.redhat.com/en/documentation/subscription_central/1-latest/configure-console_proc_sub_allocation_new_2

4. **Red Hat Subscription Central 1-latest, Enable content access on disconnected systems with manifests, retrieved 2026-11-13. [primary]**
https://docs.redhat.com/en/documentation/subscription_central/1-latest/configure-console_assembly_creating_managing_manifests_satellite_disconnected

5. **Red Hat Satellite 6.19, Installing Satellite Server in a disconnected network environment, retrieved 2026-11-13. [primary]**
https://docs.redhat.com/en/documentation/red_hat_satellite/6.19/html/installing_satellite_server_in_a_disconnected_network_environment/installing-satellite-server

6. **Red Hat Ansible Automation Platform 2.7, Obtain a manifest file, retrieved 2026-11-13. [primary]**
https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.7/install-assembly_aap_manifest_files

7. **Freedom of the Press Foundation, SecureDrop stable docs, Set Up the Secure Viewing Station, retrieved 2026-11-13. [primary]**
https://docs.securedrop.org/en/stable/admin/installation/set_up_svs.html

8. **SecureDrop latest docs, Migration from Tails, retrieved 2026-11-13 (states Qubes transition complete as of late 2026; Tails support ends 2027). [primary]**
https://docs.securedrop.org/en/latest/admin/migration/index.html

9. **freedomofpress/securedrop-workstation README, retrieved 2026-11-13. [primary]**
https://github.com/freedomofpress/securedrop-workstation

10. **SecureDrop Workstation FAQ, retrieved 2026-11-13. [primary]**
https://workstation.securedrop.org/en/latest/journalist/faq.html

11. **Offline Tools OS / Vermont Secure Computing, retrieved 2026-11-13. [primary]**
https://offlinelinux.com/

12. **rurogge/coldiron-os, v0.3.0, retrieved 2026-11-13. [primary]**
https://github.com/rurogge/coldiron-os

13. **EtchaOS, retrieved 2026-11-13. [primary]**
https://etcha.dev/etchaos/

14. **RouterKit OS, retrieved 2026-11-13. [primary]**
https://os.routerkit.net/

15. **AnduinOS / Aiursoft, retrieved 2026-11-13. [primary]**
https://www.aiursoft.com/anduinos

16. **Australian Cyber Security Centre, Fundamentals of Cross Domain Solutions, retrieved 2026-11-13. [primary]**
https://www.cyber.gov.au/publication/fundamentals-of-cross-domain-solutions

17. **NATO Information Assurance, Data Diode vs-diode, retrieved 2026-11-13. [primary]**
https://www.ia.nato.int/DocumentGenerator/repository/version/e00a6e94-bc94-45fa-ab3a-fb6f02f10752/vs-diode-2.1

18. **NATO Information Assurance, SyBard Diode v3, retrieved 2026-11-13. [primary]**
https://www.ia.nato.int/DocumentGenerator/repository/version/591fc866-87ea-4237-a90b-d9fc380d32df/SyBard-Data-Diode

19. **Oakdoor, Oakdoor 1G Diode, retrieved 2026-11-13. [primary]**
https://oakdoor.io/products/oakdoor-1g-diode

20. **(networking)**
Wikipedia, Air gap (networking), last edited 2026-11-03, retrieved 2026-11-13.
https://en.wikipedia.org/wiki/Air_gap

21. **Wikipedia, Air-gap malware, last edited 2026-11-08, retrieved 2026-11-13.**
https://en.wikipedia.org/wiki/Air-gap_malware

22. **InfoQ, The Guardian's Deep Dive into Qubes OS, May 2024, retrieved 2026-11-13.**
https://www.infoq.com/news/2024/05/the-guardian-quebes-os/

23. **VzxPLnHqr/nixos-airgapped, retrieved 2026-11-13.**
https://github.com/VzxPLnHqr/nixos-airgapped

24. **BerriJ/nixos-installer, retrieved 2026-11-13.**
https://codeberg.org/BerriJ/nixos-installer

25. **pete3n/nix-offline-iso, retrieved 2026-11-13.**
https://github.com/pete3n/nix-offline-iso