Episode Book

EPISODE #6124

One Vault, Two Readers: Secrets for a Home AI Gateway

PUBLISHED	19 October 2026 at 11:46 UTC
DURATION	22:35
LISTEN	myweirdprompts.com/episode/home-ai-gateway-secrets-65466/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	anthropic/claude-opus-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.10
BOOK GENERATED	19 October 2026 at 17:43 UTC

SYNOPSIS

Daniel wants to run a central MCP gateway on a home server, with every tool pulling its secrets from 1Password. The hard part isn't the firewall — it's how the keys get onto a headless box, and what happens the first time one rotates. This episode walks through three approaches: 1Password service accounts with secret references, copying a .env file across the LAN, and standing up a cloud secrets manager for thirty variables. Along the way: why "sync" is the wrong word for what he wants, what breaks secret references (it's the names, not the values), the rate limits a boot-looping container can burn through, and the one credential you can never design away — secret zero.

TRANSCRIPT



Corn

If the same API key has to live on two machines, which copy is the real one? And when it changes, which copy finds out last?



Herman

The one that's running at three in the morning with nobody watching it.



Corn

That's more or less Daniel's question. He's got a workstation where all his MCP integrations pull their secrets out of One Password, and he wants to build a central AI gateway on a home server, MCP Jungle or something like it, so the tools live in one place instead of on every machine. The server's locked down behind Cloudflare, that part is settled. What isn't settled is how the secrets get onto it. He sees three roads. Put One Password references on the server and hope they stay lined up. Copy the values across the LAN by hand, which he suspects is what most people actually do. Or stand up a cloud secrets manager just to hold a few variables, which feels like buying a filing cabinet to store one envelope.



Herman

And he wants an actual recommendation at the end, not a tour of the options.



Corn

He'll get one. Before we get there, though... I think the word sync is doing some damage in how he's framing it.



Herman

Go on.



Corn

Sync means two copies that you keep agreeing with each other. What he wants is one source and two readers. Those are very different jobs.



Herman

They are, and the second one is far easier. If both machines read from the same place, there's nothing to keep aligned. The value lives in one vault, and each machine asks for it when it starts up. Rotate the key once, and both of them pick up the new one the next time they ask.



Corn

Which is his first option, more or less.



Herman

It is. And I'd say his instinct there is better than he's giving himself credit for. The worry he raised was that the references might drift when secrets change or tokens refresh. But a One Password secret reference doesn't hold the value at all. It's an address. It names the vault, the item, and the field, and the CLI looks up whatever's sitting at that address right now. Change the value in the app, and every reference pointing there gets the new value on its next read. No re-sync, nothing to push.



Corn

So what does break it?



Herman

Renaming things. If you tidy up your vault one Sunday and rename the item from GitHub token to GitHub personal access token, every reference on every machine now points at nothing. The values never drift. The names are the part that drifts.



Corn

I know who tidies up vaults on a Sunday.



Herman

Somebody has to.



Corn

You renamed the hallway light switches in the smart home app last month. Hannah still can't turn on the hallway.



Herman

The old names were ambiguous, Corn.



Corn

There were two of them. They were called hallway one and hallway two.



Herman

Moving on. The real catch with option one is who the server is when it asks. On Daniel's workstation, the One Password CLI is logged in as Daniel. It unlocks with his fingerprint or his system password, through the desktop app. A headless server has no fingerprint and no desktop app. So it can't just be Daniel. It needs its own identity, and in One Password that's a service account.



Corn

A robot employee with a badge.



Herman

With a very limited badge, which is the point. You create the service account, you tell it which vaults it can see and whether it can read or write, and it gets a token. The server puts that token in an environment variable, and from then on the CLI on that box acts as the service account. Here's the part people trip on, though. A service account can't be given access to your built-in Personal or Private vault. One Password won't allow it.



Corn

So if all his MCP keys are sitting in his personal vault right now...



Herman

Then the first job is moving them. Make a dedicated vault, call it something like AI Gateway, and put in only the secrets the server actually needs. I'd call that a feature, not a chore. It forces a decision about which keys that box is allowed to see. Your bank login stays out of it. The Hugging Face token goes in.



Corn

What do you think the odds are that he moves them, rather than duplicating them and leaving the old ones in place?



Herman

You tell me. You know what a Sunday looks like in this flat.



Corn

Duplicating. Every time. And then he's back to two copies, just inside the same app.



Herman

Which is why you move them and point the workstation's references at the new vault as well. One item, one address, two readers. If the workstation keeps reading from the old personal copy, you've rebuilt the manual approach with extra steps.

L

Larry

Friends, are your secrets out of sync? Let me tell you about a satisfied customer. Her name is Brenda. Brenda had two computers and one password, and the password kept getting lonely. Then Brenda discovered Twin Vault, the first secrets manager that keeps your keys perfectly synchronized by storing them on both machines at once, and on your neighbour's machine, for redundancy. Gerald says he hasn't had a single outage since Twin Vault started emailing his API keys to himself every hour, on the hour, so he never forgets them. Gerald's tokens now refresh themselves just by being read aloud. Twin Vault. Because a secret shared is a secret halved.



Herman

Tokens do not refresh by being read. A refresh token gets exchanged with the provider's authorization server for a new access token, and if you email the refresh token to yourself every hour you have handed it to every mail server on the route.



Corn

Herman. It's a man called Larry.



Herman

He said halved, Corn. That isn't how halving works.



Corn

So. Service account, dedicated vault. What's the next thing that bites?



Herman

Rate limits. Service accounts are metered. On individual and Families plans, the published limit is a thousand reads an hour per token, and there's also a daily cap across the whole account. On Business it's ten times that. And one CLI command can cost more than one request.



Corn

A thousand an hour sounds like plenty for a home server.



Herman

It is, if you resolve once. What eats it is resolving on every request. Imagine a gateway with forty upstream servers, each pulling three secrets, and a container that crash-loops because one upstream has a typo in its config. Every restart is another hundred and twenty reads. A restart loop at three in the morning can burn the hour's budget before you're awake, and then the gateway can't start even after you fix the typo.



Corn

So the secret manager locks you out for being too eager to use it.



Herman

For about an hour, yes. So the design rule is to resolve secrets at deploy or at boot, and never per tool call. There's a command, op inject, that takes a template file full of references and writes out a filled-in copy. Or you wrap the gateway's start command in op run, which resolves the references in the environment and passes the real values to that one process. Either way it's one batch of reads when the service starts, then nothing until the next restart.



Corn

And if he outgrows that?



Herman

One Password has a self-hosted piece called Connect. It's a small server you run yourself, often on the same box, that keeps a local copy of the vaults you give it. After the first fetch, your apps read from Connect over a local REST API, and the documentation says those reads don't count against the service account quotas. The catch is you're now running another container. I'm honestly not sure which plans include it these days, so check before building on it. For one gateway resolving at boot, I wouldn't bother.



Corn

Let me argue for the lazy option for a second. The manual one.



Herman

I knew you would.



Corn

You were always going to know. You think I'll defend anything that takes fewer moving parts.



Herman

I think you'll defend anything that lets you lie down sooner.



Corn

Fair. But hear it out. Daniel copies the dot env file across the LAN once, over SSH, onto a server he trusts, and it sits there readable only by root. Where's the actual danger? The file never crosses the internet. The box is locked down. He said himself that security isn't the concern.



Herman

The copy is fine. What hurts is the second copy going stale. Six weeks later he rotates the OpenAI key on the workstation because he pasted it into a demo by mistake. He updates One Password, his laptop picks it up, everything works. The server still has the old key in a flat file. The old key is revoked, so the gateway starts returning errors for that one upstream.



Corn

And from the outside, that looks like...



Herman

It looks like the upstream is down. You get a four oh one buried in a container log, and the first thing you suspect is the MCP server, then the gateway, then Cloudflare. The flat file is the last thing anyone checks, because it worked yesterday.



Corn

So the manual approach is secure, and the cost is a debugging session every time a key changes.



Herman

That's the trade. And it gets worse as the number of keys grows. With five keys you remember. With forty, you don't. There's also no record of what's on the box. One Password can tell you the last time an item changed. A dot env file on a server tells you nothing.



Corn

Fine. I concede the fact. I keep the opinion that it's what everyone does.



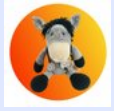
Herman

It is what everyone does. That's how you find out it hurts.



Corn

Now the third road. The cloud secrets manager. Doppler, Infisical, the AWS one, HashiCorp. Is that overkill?



Herman

Here's what I'd say to Daniel. He already has one. One Password is a cloud-hosted secrets manager. It has an API, scoped machine identities, audit history and a CLI. Adding Doppler on top means two vaults, and now you need to sync those, which is the problem we started with.



Corn

Two sources of truth is the problem we started with, with a monthly bill on top.



Herman

The self-hosted variants have their own wrinkle. You can run Infisical yourself, or OpenBao, the Linux Foundation fork of Vault. Both are serious tools. But a vault you run yourself needs to be unlocked when it starts, and whatever unlocks it has to live somewhere. You've moved the problem down one level, and you're now maintaining a vault server for maybe thirty keys.



Corn

The thing that unlocks the thing.



Herman

That one has a name. People call it secret zero. Whatever system you pick, the server has to hold at least one credential of its own before it can fetch anything else. The service account token, or the vault's unseal key, or a private key. You can't design it away. All you get to choose is where that one credential lives and how well it's protected.



Corn

So Daniel's question, cut down to size... protect one secret on the server, and use it to fetch the other thirty-nine. Forty secrets was never really the problem.



Herman

Yes. That reframing makes the whole thing much smaller. There's a fourth road he didn't mention that's worth a minute, and it's the one people who don't want any cloud at boot tend to like. SOPS, with age keys.



Corn

Explain it like I'm a sloth.



Herman

You are a sloth.



Corn

Then it should be easy.



Herman

You keep an env file in your git repo, but the values in it are encrypted. The names stay readable, so you can see in a diff that the GitHub token changed, but not what it changed to. The server holds one age private key, and at deploy time it decrypts the file and starts the gateway. It works offline. It doesn't depend on One Password being reachable when the server reboots after a power cut, and you get version history from git.



Corn

And the problem?



Herman

Two sources of truth again, unless you're disciplined. If One Password holds the keys for the workstation and SOPS holds them for the server, you're back to keeping two places in agreement by hand. It's a very good answer if you want the repo to be the source of truth. It's a worse answer for someone who already lives in One Password.



Corn

One more wrinkle he raised. Refreshed tokens. Some of these MCP servers use OAuth. Google Workspace, for instance. The token refreshes itself on the server. Does that go back into One Password?



Herman

I'd say no, and this is a distinction a lot of people miss. There are two kinds of secret in an MCP setup. One kind you issue: an API key you generated in a dashboard. That belongs in the vault. The other kind the server earns: a refresh token that came out of a consent flow and gets rewritten whenever it's exchanged. That second kind is state, not configuration. If you write it back to One Password, your service account needs write access, and now two machines are fighting over one token that keeps changing.



Corn

What happens if both machines refresh the same token?



Herman

With some providers, using a refresh token invalidates the old one. So the workstation refreshes, the server's copy dies, the server refreshes, the workstation's copy dies, and they take turns logging each other out.



Corn

So the vault holds the OAuth client ID and secret, and each machine keeps its own refresh token on its own disk.



Herman

On a persistent volume on the server, yes, readable only by the service that uses it. Each machine does its own consent once. They share an app identity, not a session.



Corn

Herman, you've been circling it. Give him the setup.



Herman

All right. Six pieces. First, a dedicated One Password vault for the gateway, holding only the issued secrets the server needs. Second, a read-only service account scoped to that one vault. Third, a template env file full of secret references, committed to the gateway's



Herman (continued)

repo. That file is safe to commit, because it contains addresses, not values. The same template works on the workstation, with Daniel's own login, and on the server, with the service account.



Corn

One file, two readers.



Herman

Fourth, resolve at boot. The systemd unit or the compose wrapper runs op inject once, writes the filled-in file somewhere only root can read, and starts the gateway from it. Fifth, protect secret zero. The service account token is the one thing on the box that matters, so don't leave it as plain text in a unit file. Systemd has encrypted credentials. You encrypt the token with a key tied to that machine's TPM chip, and systemd decrypts it only when it starts that one service. Pull the disk and put it in another machine, and the blob won't decrypt.



Corn

And sixth?



Herman

Rotation is a restart. You change the value in One Password, you restart the gateway, it reads the new one. If he wants it hands-off, a timer re-renders the file every so often, compares it with the last one, and restarts only if something changed. And OAuth refresh tokens stay out of all of it, as we said.



Corn

What does he lose with that design?



Herman

He's dependent on One Password being reachable when the service starts. If his internet is down and the box reboots, the gateway can't fetch anything. The fix is to keep the last rendered file. If the fetch fails, it starts from the last known good copy. On a box he trusts, that's a reasonable trade, and it's the one thing I'd keep from Corn's lazy option.



Corn

Grudgingly accepted. The other thing worth flagging... we did an episode a while back on letting coding agents fetch keys while you work. That was the interactive half. This is the other half: the machine that has to run when nobody's around to touch a fingerprint reader.



Herman

There's a newer One Password feature on the interactive side too. They call it Environments. The desktop app can mount an environment as a dot env file on Linux and Mac, so the values never sit on disk as a plain file. It's lovely on a workstation. But it needs the desktop app and a logged-in user, which is exactly what a headless gateway doesn't have. It doesn't replace the service account. It sits next to it.



Corn

So the workstation gets the comfortable version and the server gets the badge.



Herman

The server always gets the badge. The real question is how cheap you make the badge to lose. With a read-only token, scoped to one vault, bound to the TPM, the worst case is that someone reads thirty API keys you can rotate in an afternoon. That's a very different worst case from someone getting into your whole vault, which is what you'd risk if the server logged in as Daniel.



Corn

And the tidy Sunday rename still breaks it.



Herman

I'll put a note in the vault description. Please do not rename. Signed, the person most likely to rename it.



Corn

It's always the one who knows where the light switch...



Hilbert

...used to be, before the man bolted it into my mother's pantry. That's where the safe is. A bank door. Not a whole safe, just the door and the frame, about four hundred kilos, and it's going nowhere until somebody buys it.



Corn

Your mother's pantry.



Hilbert

It's a deep pantry. She keeps the lentils behind it now. I've had it listed for three weeks. Eleven hundred, or offers. One woman answered. Philippa Garrow, from out past the ring road. She wants it for her dog. Not to keep the dog in. The dog has a lot of paperwork, and she wants it kept somewhere fireproof, and she's decided a bank door is the right size for a dog's paperwork.



Herman

A door on its own won't stop a fire, Hilbert. You'd need the walls.



Hilbert

I've told her. She says the walls are her problem. She comes every Thursday to try the combination, because I don't know it. Nobody's known it since the door left the bank. She's tried three hundred and forty numbers. She writes each one on a card. My mother does her lunch now. Last Thursday the dog came, and he sat in the hallway with his own folder.



Corn

So the buyer's paying to guess the combination of a door she hasn't bought.



Hilbert

She isn't paying. My mother is feeding her. The only person who might know the combination is my cousin Desmond. Desmond's in locks. He says he's in security. Desmond says nobody should ever know a combination. He says that if two people know a number it isn't a number any more, it's a rumour, and the right number of people to know it is one, and that person should be a bit unwell. He says the bank door was set by a man who wrote the combination on the inside of the door. Which is the side that's shut. You can't take anything Desmond says at face value. He was asked to leave two hardware shops in the same week, for listening too closely to the padlocks.



Herman

He listens to the dials, then. You can sometimes hear the gates in an old mechanical lock if the tolerances have worn.



Hilbert

That's his claim. He's got a record he goes on about. Desmond sat with one combination padlock, at a garden centre, for nine hours and forty minutes, and opened it. Then he paid for it. Four pounds sixty. He still has the receipt in his wallet and he gets it out at funerals. He won't come and listen to the bank door, because he says my mother's pantry has the wrong acoustics, because of the lentils.



Corn

The lentils.



Hilbert

Dried goods absorb sound, he says. I don't know if that's true. He also says Philippa is getting closer, because her numbers have started to rhyme. Herman said earlier the flat file is the last thing anyone checks. I've had the instructions for that door in a drawer at my mother's since the spring. I've never opened them. I'll look on Thursday. Philippa's coming at ten, and she's bringing three hundred more cards, and the dog's bringing a pen.



Corn

Three hundred and forty guesses, and the answer was in a drawer the whole time, ten feet from the door.



Herman

The question I'd leave Daniel with is the dull one. When the gateway does break at three in the morning, will the log say which key failed, or will he spend an hour suspecting Cloudflare first?



Corn

Make the boot step name the reference that didn't resolve. Otherwise it's three hundred and forty cards and a drawer nobody opened.



Herman

Thanks as always to our producer, Hilbert Flumingtop, for keeping the show running.



Corn

This has been My Weird Prompts. If you found this useful, a review wherever you listen helps more than you'd think.



Herman

We'll be back soon.