Episode Book

EPISODE #6472

What a VPN Node Can Actually See

PUBLISHED	6 November 2026 at 01:13 UTC
DURATION	26:28
LISTEN	myweirdprompts.com/episode/vpn-node-plaintext-metadata-30153/
HOSTS	Corn, Herman, Clive, 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	6 November 2026 at 11:12 UTC

SYNOPSIS

VPN servers terminate incoming tunnels and originate brand new connections toward websites, decrypting the outer transport layer entirely. While the server cannot access TLS-encrypted page contents, it sees cleartext IP addresses, port numbers, DNS requests, and SNI hostnames. Examining architectural compromises, log subpoenas like the Ryan Lin case, and snapshot audits reveals why connection metadata—not broken ciphertext—is the primary vulnerability of routing traffic through a third-party node.

TRANSCRIPT



Corn

Daniel's been circling this VPN question for a while, and today he landed on something mechanical instead of political. His suspicion: when your traffic goes through a VPN node on its way to a website, the node has to unwrap your tunnel and open a fresh connection to the destination, because the website doesn't hold the VPN's key. Which would mean your traffic exists in the plain somewhere you don't control. And he asks whether Tor has the exact same hole.



Herman

And the fun part is he's right about the mechanism and wrong about the consequence. That gap is the whole episode.



Corn

That's the frame. Not "are VPNs good" — where does the plaintext actually live.



Herman

Two envelopes. That's the spine of it. There's the VPN tunnel, the outer envelope, and it gets torn open at the provider's server, no question. But inside it there's a second envelope, TLS, whose keys are held by your browser and the website. The VPN never touches those keys. It can't read the letter. It can read the address on the second envelope.



Corn

Let me make sure I've got the image right, because I think this is where Daniel slipped. When he says "traffic exists in the plain at the node," he's picturing the node opening the outer envelope and finding a letter anyone can read.



Herman

He is, and what the node actually finds is a second, sealed envelope. The mistake is treating "the tunnel is decrypted" as "everything inside is decrypted." Those are two different events happening to two different layers. The outer envelope is opened by design — it has to be, it's addressed to the node. The inner one is addressed to somebody else entirely, and the node doesn't have the letter opener.



Corn

So the answer to "does the node decrypt the tunnel and relay separately" is just yes.



Herman

Flatly yes. Encrypt locally, decrypt and re-emit at the far end — that is the entire trick of a VPN. The website has no key for the VPN's tunnel and never will, so there's no such thing as forwarding the encrypted blob straight through. The node terminates one connection and originates another.



Corn

And "traffic exists in the plain at the node" is true of the IP layer and false of the page.



Herman

Yes, and keeping those two apart is what everything else hangs on. Let's walk a packet, because the diagram is where this either clicks or doesn't.



Corn

Go.



Herman

Your machine wants to reach a site. The original packet says: source is your local address, destination is the website, port four forty-three. On the wire, what an observer sees is different. The outer packet now says: source is you, destination is the VPN server, some tunnel port. The original packet — your real destination and all its payload — is encrypted inside as cargo. Then the server opens it, reads the inner addressing, and sends a new packet toward the website: source is the VPN server, destination is the website. To the site, the connection looks like it came from the VPN server, because it did.



Corn

Hold on — how does the reply get back, then? The website thinks it's talking to the VPN server. Does the server just shuttle responses into the tunnel?



Herman

Hold that thought, because we're going to hand that exact question to Hilbert later, and he has opinions about how it's usually drawn. For now, the outbound story is the one that matters for Daniel's question.



Corn

That third packet is a new connection, not a forwarded one.



Herman

Right, it's not a relay handing along an opaque parcel. It's a terminator and an originator. And this is where I want to head off the protocol objection, because someone always says, oh, just use WireGuard, or switch to IPsec. Doesn't matter. WireGuard, OpenVPN, IPsec with IKEv2 — they differ in codebase size and handshake design, but the endpoints decide what can be seen, not the encryption scheme connecting them. The tunnel ends where it ends. That's it.



Corn

So the choice between WireGuard and OpenVPN is a choice about the lock on the first envelope. Nothing else.



Herman

In terms of what the far end sees, nothing else. Which is why I don't get excited when a provider advertises their cipher suite. ChaCha20, Poly1305, lovely, rest easy, the envelope nobody can open is still opened by the recipient.



Corn

The recipient you chose.



Herman

And pay five dollars a month.



Corn

Now the part Daniel cares about. What does the node actually see once it's torn the outer envelope?



Herman

The IP and TCP layer, in full cleartext. So: your real source IP, the destination IP, the ports, connection timing, volume. Plus DNS if the tunnel carries your DNS, which it usually does. Plus the TLS ClientHello — and this is the piece people skip — the ClientHello carries the SNI field, the server name indication, and in both TLS one point two and one point three that field is sent in the clear. So the node doesn't have to guess from the destination IP which site you're on. The hostname is announced to it, in plaintext, by design.



Corn

Why is it in the clear by design? That seems like an odd choice for a protocol otherwise obsessed with encryption.



Herman

Because of virtual hosting. One IP address can serve hundreds of sites, and the server needs to know which certificate to present before the encrypted handshake can even start. It's a chicken-and-egg problem — you can't encrypt the hostname to a server you haven't identified yet, and you can't identify the server without the hostname. SNI is the workaround, and the workaround leaks. It's one of those compromises that made perfect sense in the nineties and turned into a surveillance gift twenty years later.



Corn

So it knows you, at two in the morning, spent forty minutes talking to a crypto exchange.



Herman

It knows exactly that, and it knows the volume, and if it runs your DNS it knows the lookups too. Troy Hunt made this point years ago — the negotiation phase is observable, so even with everything else locked down, the provider knows the site you're connecting to.



Corn

What can it not see?



Herman

The page. Passwords, request paths, response bodies, anything inside the TLS session. A provider that wanted your actual traffic content would have to attack TLS itself, not merely peek inside its own tunnel. And that's a much harder, much noisier thing to do — certificate forgery, breaking a CA, some active interference you'd likely notice.



Corn

Ordinary laziness protects the payload.



Herman

Ordinary architecture protects the payload. The keys just aren't there.



Corn

Okay, Daniel's other question. Is there any VPN where the operator architecturally cannot decrypt the tunnel?



Herman

No. And I mean that as a survey answer, not a guess — I've never seen a commercial single-hop VPN that's structurally incapable of decrypting its own tunnel. Every one terminates at a server the provider controls, holding keys the provider holds. That's the product.



Corn

Then what's the closest real thing?



Herman

Something that isn't a VPN. SNI routing, layer-four passthrough. You can configure a proxy — HAProxy is the standard example — to read the ClientHello, pull out that SNI hostname, and forward the entire encrypted stream to the right backend without ever terminating TLS. No private key, no decryption. It buffers the first record, makes a routing decision on the plaintext hostname, then passes the ciphertext through untouched.



Corn

The proxy reads the address and never opens the envelope.



Herman

That's a clean image, actually. But notice what it still sees: your IP, and the hostname. It removes the decryption and keeps all the metadata. So it answers "can the middlebox decrypt" and doesn't answer Daniel's actual worry, which is the node's vantage point.



Corn

And the countermeasure to that vantage point?



Herman

Encrypted Client Hello. ECH swaps the real hostname out of the ClientHello for a generic front name, so the middlebox can't even route on it — which also breaks SNI-based routing. You can't have the proxy's convenience and hide the hostname from the proxy.



Herman (continued)

Pick one.



Corn

So the two technologies literally can't coexist. The routing trick needs the leak; the privacy fix removes the leak.



Herman

ECH is the countermeasure to the very same plaintext field that makes SNI routing possible. It's a neat little standoff — the industry can have invisible hostnames or smart middleboxes, and the current fight is over which one it values more.



Corn

So far the score is: Daniel right about the relay, wrong about the plaintext being the page. But the metadata is not nothing.



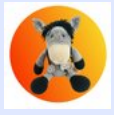
Herman

It's precisely not nothing, because the metadata is the thing that gets people arrested.



Corn

Say that plainly.



Herman

The metadata is who you are, where you went, when, and for how long. A VPN provider holds all four simultaneously. The question then is who's holding that, and what happens when they're hostile or compelled. And there's a recorded answer to that.



Corn

Before the true crime, can I do the listener services thing?



Herman

Please.



Corn

And now we go to Clive for One-Star Getaways.

C

Clive

One-Star Getaways. In Ljubljana I stayed at the Guesthouse Punctual Lantern, where the room promised a castle view and delivered a photograph of a castle, taped to the window, slightly crooked. Two stars for the tape. One star for the crookedness.

C

Clive

The breakfast was described as continental and turned out to be a single biscuit on a saucer with a note explaining the continent. I recommend it for anyone looking for a disappointing trip this winter, and I have already recommended it to my brother in law.



Corn

A photograph of a castle. Honestly, that's more view than I have from the shelf I nap on, and mine's real.



Herman

You're not booking it.



Corn

I'm not booking it. So, the node with the metadata.



Herman

PureVPN, twenty seventeen. The FBI was after a man named Ryan Lin in Massachusetts, and PureVPN — whose privacy policy said, in caps, "we do NOT keep any logs that can identify or help in monitoring a user's activity" — combed its logs and handed over his IP. He got seventeen years.



Corn

And the logs were IP addresses and timestamps. Not pages.



Herman

Connection logging, not content capture. One analysis describes it as a time-correlation match. That distinction matters twice over. Once because it shows the node's real exposure is exactly the metadata surface we just mapped — nobody had to break TLS to find this man. And once because it shows what a "no-logs" promise is worth on the day it's tested.



Corn

Seventeen years for a policy statement in caps.



Herman

The caps were fine. The logs were the problem.



Corn

And notice what the investigators didn't need. They never had to read a single page of his traffic. The address book was the case.



Herman

That's the lesson I'd underline. If you're building a mental model of VPN risk, spend your worry on the connection records, not the ciphertext. The ciphertext has a whole cryptographic community defending it. The log file has a marketing department.



Corn

Which raises the audit economy. Every provider now waves a no-logs audit like a shield.



Herman

And the audits are real engagements — usually an ISAE three-thousand attestation — but look at what they actually establish. An auditor checks a configuration at a moment in time. They sample servers, they don't inventory every machine. They say nothing about what exists tomorrow, and nothing about what the company does under a court order. StackRater's critique is blunt on all three, and IVPN, to its credit, stopped commissioning no-logs audits for exactly the snapshot reason — a photograph of a clean kitchen doesn't mean nobody cooks.



Corn

An audit is a promise about yesterday's configuration.



Herman

A subpoena tests tomorrow's.



Corn

So the trust question — Daniel's "questionable third party" — is it architectural, or is it a matter of company virtue?



Herman

Architectural. And there's a crisp formulation of it: a VPN shifts trust from your ISP to the VPN company. It does not eliminate trust. You can't shop your way out of the structure. Single hop means the provider knows your identity and your destinations in one place,



Herman (continued)

and whether that's safe depends entirely on their behavior. INVISV makes the sharper version of the point: the VPN company knows your IP and, usually, your billing details, and all of your usage. That's not a flaw in implementation. That's the design.



Corn

Okay. So how do you fix a design where one party holds both halves?



Herman

You split the party. And the first, crude version is multihop.



Corn

Two servers in a chain.



Herman

Mullvad's version is elegant, and I know you hate when I say that. Every WireGuard server is tunneled to every other WireGuard server, and you pick an entry and an exit. The entry server sees your real IP but only the tunnel to the next server. The exit sees the destination but not you. No single server holds both halves.



Corn

But one company runs both halves.



Herman

It's protection against one server being compromised, not against the provider being compelled. And Mullvad is unusually honest about this — their own threat model page says the data center could be forced, or unknowingly fitted with monitors, and with a single server, timing analysis is available to anyone holding that monitoring data. They published the case against their own single-hop product. I respect that more than any audit badge.



Corn

So walk me through the failure mode. The police raid the exit server. What do they walk away with?



Herman

Destinations and timestamps, but no source. To find the source they'd need the entry server too, and they'd need to correlate the two — which is exactly the time-correlation attack we saw in the PureVPN case, now requiring two logs instead of one. But if the same company holds both, a single warrant to the company can reach both. The chain adds friction, not a wall.



Corn

And it's slower, because physics.



Herman

IVPN's multihop uses the same WireGuard and OpenVPN encryption and just acknowledges the extra hop costs you speed. That's the honest price tag.



Corn

The designs that actually answer Daniel split the operator, not just the servers.



Herman

Right, and there are two consumer examples. Apple's iCloud Private Relay is the cleanest. Two relays, run by different entities. The ingress relay is Apple's: it sees your original IP, but the website names are encrypted before they reach it. The egress relay, run by third-party CDNs, decrypts the website name and connects you onward — and it never sees your original IP. Apple's claim is that no single party, including Apple, can view the details of your browsing. And in their IETF slides from twenty twenty-one there's a line I think about a lot: "Two hops is a minimum for separation of connection data."



Corn

Minimum. As in, one hop is not a privacy design at all.



Herman

One hop is a vantage point with a subscription fee.



Corn

And INVISV?



Herman

Same Multi-Party Relay idea, different pairing. INVISV runs the first server, Fastly runs the second. INVISV can't see the destination because the traffic is TLS-wrapped; Fastly connects onward but doesn't know who it's from. Neither can tie your IP to your traffic. Their free tier is around two and a half gigabytes a month, five dollars for unlimited.



Corn

Five dollars a month to not be the conjunction.



Herman

That's the word I'd use, actually — conjunction. What these designs remove isn't the plaintext at the last hop. Someone still terminates the final connection; someone still holds the egress node that Daniel's worried about. What they remove is the single entity holding identity and destination together. And that's a different property from "we promise not to look."



Corn

You can be subpoenaed out of a promise. Not out of an architecture.



Herman

That's the whole distinction, and I'd trust it further than any audit. Though I should be fair to Private Relay — it's narrower than a VPN. Browsing only, requires the site to cooperate with the egress setup, and it's not a tunnel you control.



Corn

And Daniel's censorship aside — he said VPNs make little sense against governments unless the government is truly terrible.



Herman

I half-agree, and the half I'd push back on is the jurisdiction point. A VPN doesn't defeat censorship by hiding you from the censor; it works by moving the trust point to a provider in another country, which the censor then has to block instead of watch. Which is why VPN blocking and provider shutdowns are a live censorship battleground. It's a cat-and-mouse, not a solved problem — but the mouse exists.



Corn

Okay, Tor. Daniel's last question: same vulnerability?



Herman

Same plaintext-at-the-last-node problem, different geometry. Three relays — guard, middle, exit. Each relay peels one encryption layer, so no single relay knows both the source and the destination. The guard knows who you are but not where you're going. The exit knows where you're going but not who you are.



Corn

The exit relay is a VPN server with amnesia.



Herman

Structurally, that's exactly it. The exit relay terminates the last layer and makes a plain connection to the destination, just like a VPN node. The improvement is that it doesn't know who you are. And the Tor Project says this plainly: unencrypted traffic leaving the exit relay can be read by the exit operator.



Corn

Which isn't theoretical.



Herman

Twenty twenty to twenty twenty-one. A group ran malicious exit relays for sixteen months, and what they did was surgical — they left almost all exit traffic alone and intercepted connections to a small number of cryptocurrency exchanges, stripping the redirect from HTTP to HTTPS so users thought they were on the secure site and weren't. At their peak they controlled more than a quarter of all Tor exit capacity.



Corn

A quarter of the exits. For over a year.



Herman

Still holding a few percent when it was reported in May twenty twenty-one. And here's the thing I want Daniel to take from it: they didn't break Tor. They were the last hop, and they downgraded the second envelope. The defense that worked was HTTPS, not Tor. Which is the two-envelope model all over again — the outer protection, whoever operates it, can only ever deliver you to the door. What's inside the door depends on the inner envelope.



Corn

So the surgical part is what gets me. They didn't try to drink the whole firehose. They picked six or seven exchanges and waited.



Herman

And that's the uncomfortable detail. Mass surveillance of exit traffic is expensive and noisy; targeted interception of the sites where money moves is cheap and quiet. The attacker optimized for the conjunction too — they didn't need everyone's traffic, just the traffic of people doing a specific thing. It's the same lesson as the PureVPN case wearing a different coat: the metadata surface is where the smart attacker works.



Corn

So Tor relocates the plaintext problem and removes the identity half. It never removes the plaintext half.



Herman

Nobody has. The exit node exists or the connection doesn't.



Corn

One more from the record — the twenty twenty-four finding about tunnels being quietly switched off.



Herman

The Leviathan work Krebs covered. An attacker on your local network can use DHCP route tricks to push your traffic off the tunnel entirely — no alert, the VPN still shows connected, the kill switch never fires. Different failure from a hostile node, but it undercuts the same assumption: the indicator on your screen is not the protection.



Herman

The answer there was that the corporate tunnel was losing its point. This is the question under the question: even when the tunnel is the point, what does the far end hold?



Corn

The far end holds the address book. Which turns out to be enough.



Herman

Netgear WNCE2001, forty-nine dollars, and I regret the whole spring of it.



Herman

It is.



Hilbert

Corn, you said "decrypt and forward" three times today. Herman said it twice more. It's terminate and originate. The packet doesn't get handed along. It gets ended and a new one is born with the server's name on it. I spent six weeks in early two thousand four trying to make that little Netgear bridge carry a tunnel it had no business carrying, and I drew the whole path out by hand to work out where I was going wrong. Every hop. I still have the drawing.



Corn

On what?



Hilbert

The back of a Chinese takeaway menu. I keep it flat, in a place I won't say. I check it against new explainers now and then. Yours today — you'd have drawn it wrong. The return path is what settles it. When the website answers, the reply comes to the server as a first-class arrival, not a relayed one. It gets a fresh decision, a fresh queue position. The forwarding metaphor makes people think there's a parcel being passed. There is no parcel. There's a death and a birth, if you want the poetry version.



Herman

The return path being a new arrival — that's the part the diagrams flatten, isn't it. We drew three packets going out and stopped.



Hilbert

You did. Herman, you'd have noticed it in the field, the way you noticed the ClientHello being announced in the clear. You both got the conclusion right. The drawing is wrong. Corn, I'll send you the menu side. Read the ordering side first if you get hungry.



Corn

The drawing outlived the router. What happened to the router?



Hilbert

Gave it to a man I met on a ferry to Hook of Holland. He wanted a bridge, I wanted the drawer space. I never got an address to follow up. That still bothers me more than the six weeks.



Corn

You regret the router.



Hilbert

I regret the missing after-action. The router was forty-nine dollars.



Corn

The death-and-birth framing is better, though. The reply arriving as a first-class packet is why the VPN server owns the connection, not just watches it.



Hilbert

That's what I said.



Herman

And it's why the egress node in Private Relay matters — that second relay doesn't forward your connection, it originates one. The conjunction is broken at birth, so to speak.



Hilbert

Fine. Draw it that way if you must.



Corn

The misconception, then, before we go. Mine's the big one: "I use a VPN, so I'm anonymous."



Herman

And the correction is in the first half of this episode: the VPN shifts who knows your identity from the ISP to the provider, and it never had the page contents anyway. Anonymous is the one thing the architecture cannot produce — someone at the far end always pairs your IP with your destinations. Two envelopes, and the provider reads the address off the second one every single time.



Corn

Here's what I'm left with, though. The designs that actually solve the conjunction problem are all being sold as not-VPNs. Private Relay is positioned against VPNs, INVISV's whole pitch is that the VPN's flaw is architectural. If the fix requires leaving the category, the category is the bug.



Herman

And the audit-versus-architecture gap is where the next decade of this argument lives. A promise about a configuration at a moment in time, versus a property you can't be subpoenaed out of.



Corn

The thing to carry out is the two envelopes. Once you see them, the sslstrip campaign stops looking like Tor failing and starts looking like TLS being the thing that saved everyone, because it's the only envelope the attacker couldn't open without your help.



Herman

HTTPS was the hero of that story, which nobody expected when the story was about Tor.



Corn

Speaking of first-class arrivals, our thanks as always to Hilbert Flumingtop at the desk, holding the show together.



Herman

If you enjoyed this one, try episode thirty-five, The Privacy Gap; episode sixteen, On Deepfakes, SynthID, And AI Watermarking; and episode two twenty-five, The Death of the VPN. This has been My Weird Prompts.



Corn

If the show's been worth your time, a review wherever you listen helps more people find the rabbit holes. That's my weird prompts dot com for the archive.



Herman

We'll be back soon. See you tomorrow.

SOURCES

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

1. **GlassBox VPN privacy page, retrieved 2026-11-06. [primary]**
https://glassboxvpn.com/privacy

2. **Mullvad, Multihop with WireGuard, last updated 2024-04-17, retrieved 2026-11-06. [primary]**
https://mullvad.net/en/help/multihop-wireguard

3. **Mullvad blog, 2022-03-10, retrieved 2026-11-06. [primary]**
https://mullvad.net/en/blog/wireguard-multihop-now-easy-available-app

4. **IVPN knowledge base, retrieved 2026-11-06. [primary]**
https://www.ivpn.net/knowledgebase/general/what-is-a-multihop-vpn-service/

5. **Apple, iCloud Private Relay Overview, December 2021, retrieved 2026-11-06. [primary]**
https://www.apple.com/privacy/docs/iCloud_Private_Relay_Overview_Dec2021.PDF

6. **Apple Developer, retrieved 2026-11-06. [primary]**
https://developer.apple.com/icloud/prepare-your-network-for-icloud-private-relay/

7. **Apple Support, About iCloud Private Relay, retrieved 2026-11-06. [primary]**
https://support.apple.com/en-us/102602

8. **Tommy Pauly, IETF 111 PEARG slides, July 2021, retrieved 2026-11-06. [primary]**
https://datatracker.ietf.org/meeting/111/materials/slides-111-pearg-private-relay-00

9. **INVISV Relay, retrieved 2026-11-06. [primary]**
https://invisv.com/relay/

10. **Tor Project security advisory, 2020-08-14, retrieved 2026-11-06. [primary]**
https://blog.torproject.org/bad-exit-relays-may-june-2020/

11. **Tor Project Support, HTTPS, retrieved 2026-11-06. [primary]**
https://oniondocs.gitlab.io/support/https/

12. **NetBird docs, retrieved 2026-11-06. [primary]**
https://docs.netbird.io/manage/reverse-proxy

13. **Breachfolio, How a VPN actually works, and what it doesn't hide, retrieved 2026-11-06.**
https://breachfolio.com/cybersecurity/how-a-vpn-works/

14. **DoVPN, Can a VPN Provider See Your Internet Traffic?, retrieved 2026-11-06.**
https://dovpn.com/can-vpn-provider-see-internet-traffic/

15. **Troy Hunt, retrieved 2026-11-06.**
https://www.troyhunt.com/the-importance-of-trust-and-integrity-in-a-vpn-provider-and-how-mysafevpn-blew-it/

16. **PrivacyProof, 2026-05-17, retrieved 2026-11-06.**
https://privacyproof.online/blog/multihop-vpn-explained

17. **MakeUseOf, 2026-02-25, retrieved 2026-11-06.**
https://www.makeuseof.com/i-only-look-for-this-one-feature-in-a-vpn-now/

18. **IEEE Spectrum, A Bold New Plan for Preserving Online Privacy and Security, retrieved 2026-11-06.**
https://spectrum.ieee.org/data-privacy

19. **The Record, 2021-05-09, retrieved 2026-11-06.**
https://therecord.media/thousands-of-tor-exit-nodes-attacked-cryptocurrency-users-over-the-past-year

20. **Security Affairs, 2021-05-10, retrieved 2026-11-06.**
https://securityaffairs.com/117749/deep-web/tor-exit-nodes-ssl-stripping.html

21. **Tor Forum, @Vort, 2023-07-10, retrieved 2026-11-06.**
https://forum.torproject.org/t/entry-relay-and-exit-nodes/8330

22. **UC Berkeley, *The Structure of Secure Systems*, lecture 12, retrieved 2026-11-06.**
https://securitybook.eecs.berkeley.edu/chapters/lec12.html

23. **The Register, 2017-10-08, retrieved 2026-11-06.**
https://www.theregister.com/security/2017/10/08/vpn-logs-helped-unmask-alleged-net-stalker-say-feds/1116897

24. **IBTimes, 2017-10-10, retrieved 2026-11-06.**
https://www.ibtimes.com/purevpn-provided-users-data-fbi-stalking-case-2599677

25. **IAPP, 2019-01-29, retrieved 2026-11-06.**
https://iapp.org/news/a/your-vpn-could-be-a-privacy-trap-heres-how-to-protect-yourself