Address Poisoning is a Wallet Bug, Not a User's Fault

Here are two Ethereum addresses. One is real. One was planted by an attacker to steal your money.

```
0xd28b65...6922
0xd28b65...6922
```

Can you spot the fake? You've got three seconds. That's about how long most people look before hitting Send.


Spot the fake

You know those "spot the difference" games from kids' magazines? This is the same thing, except getting it wrong costs you $68 million. That's not hypothetical. That's what someone actually lost.

Your wallet is helping the attacker

Let's be honest about what's happening here. Your wallet shows you `0xd28b...6922` — six characters at the front, four at the back — and calls that a security feature. It's not. It's the opposite. It's the thing that makes the attack work.

An attacker sends your wallet a zero-value transaction from a lookalike address. Your wallet logs it in your history. Next time you go to send funds, you copy the address from your recent transactions. Why wouldn't you? It looks identical.


Wallet history

The wallet truncated the address to be "helpful." The attacker matched the visible parts. The wallet showed both entries the same way. You picked the wrong one. And somehow this is *your* fault?

Stop telling people to "verify their address"

I keep seeing this advice. "Always double-check the full address before sending." Double-check WHAT exactly? Forty random hexadecimal characters? Go ahead, try it:

`0xd28b651093A1288fDE8C08A8B6A847dFF3519b6922`

Nobody can meaningfully verify that. It's not a human-readable string. It's line noise. Telling people to "carefully check" a 40-character hex string is like telling drivers to memorize every license plate they see on the highway. At 70 miles per hour. While also steering.


Blame the user

The "just verify" crowd has it backwards. If your interface requires superhuman attention to avoid a catastrophic mistake, the interface is broken. Not the user.

Some protocols figured this out years ago

Here's the part that makes me angry. This isn't an unsolved problem.

Uniswap's router? `0x000000000022D473030F116dDEE9F6B43aC78BA3`. ENS? `0x00000000000C2E074eC69A0dFb2997BA6C7d2e1e`. Seaport? `0x0000000000000ADc04C56Bf30aC9d3c0aAF14dC`.

See all those leading zeros? These are *addresses with long leading zeros* — mined deliberately by spending GPU time upfront. An attacker trying to spoof one of those would need to match all those zeros *plus* enough of the remaining characters. The cost scales exponentially: every additional leading zero nibble multiplies the difficulty by 16×. Pocket change becomes decades of compute time.


Protocols with leading zeros

They didn't wait for a committee. They didn't write a proposal and wait two years for review. They just did it. And it works.

So why isn't everyone doing this?

Three things were blocking it

The hardware problem. Mining a pretty address used to mean renting GPU time on a farm or owning serious hardware. Regular users sending $500 to a friend don't have access to that. Shouldn't need to.

The expertise problem. Even if you wanted to build a tool for this, you'd need deep knowledge of GPU programming — CUDA for NVIDIA, Metal for Apple, OpenCL for AMD. Each platform is its own world with its own toolchain, its own quirks, its own debugging nightmares. Build it for one and everyone else is locked out. The handful of tools that existed were command-line utilities that required you to install specific drivers, configure your GPU compute environment, and pray everything linked correctly. That's a developer task, not a user task. Adoption was basically zero outside of protocol teams with GPU expertise on staff.

The display problem. Say you go through all that trouble and actually mine an address with long leading zeros. Great. Now open your wallet. It shows `0x0000...78BA`. The zeros that protect you? Truncated away. The wallet strips out the exact feature that makes the address safe. There was no standard way to display an address with long leading zeros that preserves its security properties.


Three barriers

Three real problems. All three are now solved.

The fix exists. Ship it.

WebGPU killed the hardware and expertise barriers in one shot. It runs in the browser. Any browser. Any GPU — NVIDIA, AMD, Apple Silicon, whatever you've got. No drivers to install. No toolchains to configure. No CUDA expertise required. PrettySafe.xyz uses WebGPU to mine an address with 9 leading zeros in a few minutes on a normal laptop. Open a tab, click a button, wait. That's it.

ERC-8117 killed the display problem. Instead of showing `0x00000000000C2E074eC...` (which wallets truncate into uselessness) or `0x0000...78BA` (which hides the protection), ERC-8117 defines a compact notation: `0x0₁₁C2E...` (subscript) or `0x0(11)C2E...` (ASCII fallback). The subscript tells you there are 11 leading zeros. You can see the protection at a glance. No squinting at 40 characters. No trusting truncation.

 ERC-8117 before and after

And the math is wild. One minute of your laptop's GPU time creates an address that would cost an attacker 32 years to fake. One minute versus 32 years. The asymmetry is staggering.

So what are we waiting for?

No new protocol is needed. No hard fork. No governance vote. No committee approval. Every piece is ready today: WebGPU works in Chrome, Firefox, Safari, and Edge. ERC-8117 is a display standard — it doesn't touch consensus. Mining a safe address takes minutes.

If your wallet still shows `0xd28b...6922` in 2026, it's not protecting you. It's a phishing tool with a nice UI.

 WebGPU cross-platform

 Call to action

If you're a user: go to PrettySafe.xyz and mine yourself a safe address. It takes minutes. Your money deserves a fingerprint, not a blur. And if your wallet doesn't support ERC-8117 display yet, tell them. Loudly. Switch to one that does.

If you build wallets: implement ERC-8117 display. The spec is done. The code is trivial. Better yet, bake WebGPU mining into your onboarding — new users should get safe addresses by default, not addresses that look like every other address on the network. You've been shipping truncated hex strings as your address display for years. That's the UI the attackers need. Stop giving it to them.