FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

WHM Is IP-Locked, So Why Can't I Reach It? — Cloudflare Proxied Hostnames vs Origin IPs

A field note about the day my own firewall rule locked me out of WHM — a rule that was correct, on a hostname that was orange — and the read-only checks I now run before I touch a single admin allowlist.

Daanial Farrukh · 06 November 2026 · https://quabin.com/whm-ip-locked-cloudflare-proxied-hostname/

This one has no customer quote at the start, because the person who got it wrong was me.

A cPanel server was sitting behind a distributed firewall, the kind that is enforced outside the VM. The job was simple and sensible: stop the world from reaching the admin ports. SSH on 22 and WHM on 2086 and 2087 would answer one source only — my admin IP.

So I wrote the rules the way you are supposed to write them. Allow from the admin IP. Then an explicit drop from any, placed above the catch-all allow at the bottom of the policy. The firewall reads top-down and the first match wins, so the order was doing real work.

SSH still worked. WHM did not.

My IP had not changed. The rule was enabled. The order was right. I had just locked myself out with a rule I could read line by line and find nothing wrong with.

There was nothing wrong with it.

That's the trap. **An IP allowlist does not check who you are. It checks who connected last.** And when your hostname is orange in Cloudflare, the last one to connect was not you.

What Actually Happens to a Request on Port 2087

Most people think of Cloudflare as a web thing. Port 80, port 443, done. It isn't.

Cloudflare's proxy also listens on a published list of other ports, and that list includes the whole cPanel family:

- 2082 and 2083 — cPanel
- 2086 and 2087 — WHM
- 2095 and 2096 — Webmail

So a proxied hostname does not fail on 2087. It works, and that is the problem. Your browser reaches Cloudflare, and Cloudflare opens a brand-new TCP connection to your origin:

Admin laptop → Cloudflare edge → Firewall → WHM

There are two connections here, not one. The first one is yours. The second one belongs to Cloudflare. **The firewall in front of the origin only ever sees the second one.** Its source address is a Cloudflare address, so the walk down the policy goes like this:

Source = Cloudflare IP → Allow admin IP? No → Drop from any? Yes → Dropped

The rule did exactly what I told it to do. It dropped a connection from a stranger. The stranger just happened to be carrying my request.

Here is the part that trips people up: Cloudflare does pass your real address along, in the CF-Connecting-IP header. That is no help at this layer. A network firewall decides on the TCP source address before any HTTP header exists. The header is inside a connection the firewall never lets start.

Why SSH Tells You Nothing — and Why That Helps

Port 22 is not on Cloudflare's normal proxy list. So SSH to the origin IP goes straight to the server, and the firewall sees your real address. (Point SSH at an orange hostname and it simply fails, because Cloudflare's normal proxy does not carry it. That is a different symptom.)

That difference misleads you first. "SSH works, so my IP is allowed, so the firewall is fine." You go looking in WHM, in cPHulk, in the service status.

Then it helps you. Same machine, same firewall, same allowlist, same laptop. One port works and the other does not. The rule cannot be the difference, because both ports use the same rule logic. **The difference is the path.** When SSH works and WHM does not, stop reading the firewall and start asking which IP WHM's traffic is really coming from.

Why You Never See This in Testing

Three things hide it.

First, the rule is usually tested with the wrong URL. Many admins test against the raw IP while building the rule, it passes, and then their bookmark goes to the friendly hostname from then on.

Second, the hostname may have turned orange long after the firewall was built. Somebody importing a zone into Cloudflare, or proxying everything to be safe, can flip it months later. The firewall did not change. The DNS did.

Third, the error points away from the cause. A dropped connection gets no answer, so a browser going through Cloudflare should show Cloudflare's own 522 connection-timed-out page, not anything from your firewall or from cPanel. If the firewall rejected instead of dropped, the same mistake would normally show up as a 521. Either way, the page tells you the origin is unwell. The origin is fine.

How to Actually Fix This

Everything up to Step 3 is read-only.

Step 1: Find out what your hostname really resolves to.

```
dig +short server.example.com
```

If you get your origin IP back, the hostname is grey and this article is not your problem. If you get addresses that are not yours, usually starting 104. or 172.67., you are proxied. Compare against the list Cloudflare publishes:

```
curl -s https://www.cloudflare.com/ips-v4
```

Step 2: Ask the port who answered.

```
curl -skI https://server.example.com:2087/
```

Look at the server: line. If it says cloudflare and there is a cf-ray: header, then whatever you are talking to on 2087, it is not your WHM. A status of 522 here is the whole diagnosis in one line.

Step 3: Bypass Cloudflare for one request and compare.

This forces curl to the origin IP while keeping the correct hostname for TLS. It changes nothing anywhere.

```
curl -skI --resolve server.example.com:2087:203.0.113.10 https://server.example.com:2087/
```

Run it from the admin IP. Now the server line should name cpsrvd, which is cPanel's own service. Same hostname, same port, same laptop — and it answers. That is your proof: the rule allows you, and only the proxied path is blocked.

If this one fails too, the firewall or the service really is the problem, and you should go back to the rule. That result is useful as well.

Step 4: Make the admin hostname grey. (This changes state.)

In Cloudflare, go to DNS → Records, find the server hostname, and set Proxy status to DNS only. Leave the public websites orange. They need the proxy and they are not the ones being locked.

This matters even if you plan to browse to the IP. When I open the origin IP on port 2087, WHM redirects me to the server hostname on 2087. If that hostname is orange, the redirect walks me straight back into Cloudflare and into the drop rule. **The IP gets you to the door. The grey hostname keeps you there.**

After the change, wait for the old TTL to expire, then repeat Step 1. You should see the origin IP. Then bookmark the direct path:

```
https://203.0.113.10:2087
```

Step 5: Verify from both sides.

This step is where most tutorials stop and where mine starts. Working for you proves half of it. The other half is failing for everyone else.

From the admin IP:

```
curl -skI https://server.example.com:2087/
```

Expect the cpsrvd server line and no cf-ray header.

Then, from anywhere else — a phone hotspot is fine:

```
curl -skI --connect-timeout 10 https://203.0.113.10:2087/
```

Expect a timeout and nothing else. curl exits with code 28.

This error is not a bug. It is the verification. If that second command returns a login page, your lock is not locking.

On the server itself, one more read-only line shows which address WHM thinks you are:

```
tail -n 5 /usr/local/cpanel/logs/access_log
```

The first field of each line is the client. It should be your admin IP.

One warning from the way this firewall works: when the drop happens outside the VM, a capture inside the VM shows nothing at all. An empty tcpdump on port 2087 does not mean no traffic was sent. It means the traffic died before it arrived. Use the hit counters on the firewall rule for that view.

The Fix You Must Never Make

Sooner or later somebody says it: "Just add Cloudflare's IP ranges to the WHM allow rule."

It will work. WHM will load. Please don't.

Those ranges are not your ranges. They are shared by every Cloudflare customer on the planet. Anyone can add a domain to a Cloudflare account, create a proxied record pointing at your origin IP, and send their browser through it on 2087. The connection then arrives from an address you allowed. **You did not narrow your admin port. You opened it to the world with one extra step.**

The same goes for SSH. Whichever port you lock, the allow rule names your addresses, and no one else's.

If your admin IP changes often, the answer is a VPN or a jump host with a fixed address. If you truly want WHM behind Cloudflare, that is a different design: Cloudflare Tunnel with Access in front of it, and nothing listening to the public internet. It is a good design. It is not a one-line allow rule.

The Honest Trade-Off

A grey hostname publishes your origin IP in public DNS. Some people will object to that, and they are not wrong to care.

On a cPanel server the mail hostnames usually publish that IP anyway, because mail cannot go through Cloudflare's normal proxy. Hiding the origin was never going to be the control here. **The control is the drop rule.** A published IP with a closed port is safe. A hidden IP with an open port is only safe until someone finds it.

The Checklist I Run Now

1. Is the admin hostname grey? dig +short server.example.com
2. Who answers on the admin port? curl -skI https://server.example.com:2087/

3. Does the bypass work from the admin IP? curl -skI --resolve server.example.com:2087:203.0.113.10 https://server.example.com:2087/
4. Does the origin refuse a stranger? curl -skI --connect-timeout 10 https://203.0.113.10:2087/ from another network.
5. Is the drop rule above the catch-all allow? Read the policy top-down, and check the hit counter is rising.
6. Is 443 a side door? cPanel's proxy subdomains can carry whm.example.com over ordinary 80 and 443, which a rule on 2087 does not touch. grep proxysubdomains /var/cpanel/cpanel.config
7. Is anything else on the same list? Check 2082, 2083, 2086, 2095 and 2096 the same way, on every hostname that points at the box.

Plain takeaway: lock the port, then prove it from outside, every time.

What I Tell Every Customer Now

When someone tells me their admin panel is locked to one IP, I don't ask to see the rule. I ask this:

"From a network that is not on the list, what does port 2087 on your origin IP do — and is your admin hostname orange or grey?"

If the answer is "it times out, and it's grey," they're in good shape. If the answer is "I think so, the rule is in there," we're not done yet.

I write about the questions I ask in real customer rooms — the ones that change architectures and surface the problems no one else noticed. If you found this useful, drop your email below and I'll send the next one when it's ready.

FAQs

Why can't I reach WHM after locking it to my IP in the firewall?

If your WHM hostname is proxied through Cloudflare, the origin firewall sees a Cloudflare address, not yours. The allow rule misses and the drop rule fires. Check with dig +short on the hostname: if it does not return your origin IP, set the record to DNS only in Cloudflare or reach WHM through the origin IP.

Does Cloudflare proxy port 2087?

Yes. Cloudflare's proxy supports a published list of HTTP and HTTPS ports, and it includes the cPanel and WHM ports 2082, 2083, 2086, 2087, 2095 and 2096. A request to an orange-cloud hostname on 2087 reaches Cloudflare first, and Cloudflare opens its own connection to your server.

Should I allow Cloudflare IP ranges to reach WHM?

No. Those ranges are shared by every Cloudflare customer, so anyone can create a proxied record pointing at your origin and reach your WHM login from an allowed address. Keep WHM limited to your own admin IP, and use a VPN, a jump host or Cloudflare Tunnel with Access if your address changes.

Why does SSH work when WHM does not?

SSH on port 22 is not carried by Cloudflare's normal proxy, so a session to the origin IP arrives directly and the firewall sees your real address. That makes the mismatch a useful clue: the same allowlist passes one port and blocks the other, so the rule is not the cause. The path taken by WHM traffic is.

Why do I get a Cloudflare 522 error on port 2087?

A 522 means Cloudflare could not complete a TCP connection to your origin. On an IP-locked server, that usually means your firewall is silently dropping Cloudflare's connection, which is what the rule was written to do. Your server is healthy; only the proxied path is blocked.

Does a DNS-only hostname expose my server's IP?

Yes, the origin IP becomes visible in public DNS. On a cPanel server, the mail hostnames usually show it anyway. Protection should come from the firewall rule that closes the admin ports, not from keeping the address secret.

Read online: https://quabin.com/whm-ip-locked-cloudflare-proxied-hostname/