

FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

Every Email Rejected at Once — The Open-Resolver Mistake That Makes Spamhaus Answer "No"

A presales war story about a mail server that suddenly rejected every sender on the internet — Gmail, Outlook, the bank — and the two DNS queries I now run on every mail server before anyone blames the blacklist.

Daanial Farrukh · 26 October 2026 · <https://quabin.com/spamhaus-open-resolver-all-mail-rejected/>

The message came in early in the morning.

"Nobody can email us. Gmail bounces. Outlook bounces. Even the bank statements are not arriving."

The server was a customer's cPanel/WHM box on Ubuntu. It had been running mail for years without much drama. Nothing had been deployed. Nobody had touched the mail settings.

The bounce text was the same for every sender:

`is in an RBL: Error: open resolver`

The first reaction on the call was the natural one.

"So Gmail is on a blacklist?"

I asked one question back.

"All of Gmail, all of Outlook and your bank — on the same blacklist, on the same morning?"

Silence. Then: "That doesn't sound right, does it?"

It didn't. When every sender on the internet is "listed" at once, the senders are not the problem. The thing doing the checking is.

The Trap, Named

That's the trap. An RBL rejection reads like a verdict on the sender. So people look at the sender. They open the blacklist websites, paste in Google's IP addresses, and find nothing.

The error was not about the senders at all. It was about how this server asked the question.

And the answer to that was hiding in one file that nobody had looked at in years.

What "Open Resolver" Actually Means Here

A blacklist like Spamhaus ZEN is queried over DNS. When mail arrives from an address, the server reverses the IP and asks Spamhaus a DNS question about it. The flow looks like this:

Sender connects → Exim takes the sender's IP → local resolver → Spamhaus DNS → answer → accept or reject

If the answer is empty, the address is clean. If the answer is an address in 127.0.0.x, the address is listed and Exim rejects the mail.

Here is the part that trips people up. **Spamhaus does not answer queries that come through big public resolvers** such as 1.1.1.1 or 8.8.8.8. Those resolvers carry traffic for millions of users, so Spamhaus blocks them. Instead of a real answer, it returns a special code: 127.255.255.254. That code means "you asked through a public resolver, I will not tell you".

Now look at that code again. It starts with 127.. A mail server that treats any 127.x answer as "listed" now sees every single sender as listed. Gmail, Outlook, the bank, the customer's own staff at home — all rejected.

That is what "open resolver" in the bounce means. Not "the sender is an open resolver". It means **"the server asked me through a public resolver, so I refused to answer"**.

This setup is common in three places:

- Servers where someone "fixed slow DNS" once by forwarding to public resolvers.
- Images or templates that ship a resolver config pointing at 1.1.1.1 or 8.8.8.8.
- Networks where outbound DNS was once blocked, and forwarding was the quick workaround.

Why Nobody Saw It

The obvious check is `/etc/resolv.conf`. On this server it looked perfect:

```
nameserver 127.0.0.1
```

That line says "use the local resolver". Most people stop there. Local resolver, so it must do its own recursion. Green tick.

It didn't. The local resolver was Unbound, and Unbound has its own config. Inside `/etc/unbound/unbound.conf` there was a block like this:

```
forward-zone:
  name: "."
  forward-addr: 1.1.1.1
  forward-addr: 8.8.8.8
```

name: "." means "for every domain on the internet". So Unbound was not resolving anything itself. It took every question, including the Spamhaus ones, and passed it to Cloudflare and Google DNS. From Spamhaus's point of view, the question came from a public resolver.

resolv.conf told the truth about the first hop. It said nothing about the second.

Why did it suddenly start that morning, and not years ago? I never proved the exact moment the block started to bite, so I won't pretend I know. What I can say is that the config was wrong all along, and the

safe design does not depend on a third party's tolerance.

How to Actually Fix This

Everything below is read-only until Step 3, which I mark clearly.

Step 1: Prove the RBL lookups are broken — not the senders.

Spamhaus keeps two test entries for exactly this. 127.0.0.2 is always listed. 127.0.0.1 is never listed. Ask the server's own resolver about both:

```
dig +short 2.0.0.127.zen.spamhaus.org
```

```
dig +short 1.0.0.127.zen.spamhaus.org
```

A healthy server returns one or more 127.0.0.x addresses for the first query and nothing at all for the second.

A broken server gives the same answer to both. If that answer is 127.255.255.254, you have your cause: Spamhaus sees your queries coming from a public resolver.

This error is not a bug. It is the verification. The test entries exist so you can tell "my resolver is wrong" apart from "the sender is listed" in ten seconds.

Step 2: Find where the forwarding lives.

First confirm the first hop:

```
cat /etc/resolv.conf
```

If it points to 127.0.0.1, the local resolver is in charge. Now look inside it:

```
grep -rn "forward" /etc/unbound/
```

Any forward-zone: with name: "." and public addresses is the problem. If the server uses a different local resolver, look for the same idea there: a forwarder that sends everything to 1.1.1.1, 8.8.8.8 or your provider's shared resolvers.

Step 3 (changes state): Remove the forwarder.

Back up the file, then comment out the whole forward-zone: block — all four lines, including name: ".":

```
cp /etc/unbound/unbound.conf /etc/unbound/unbound.conf.bak
```

```
#forward-zone:
#   name: "."
#   forward-addr: 1.1.1.1
#   forward-addr: 8.8.8.8
```

With no forwarder, Unbound does full recursion itself, starting from the root servers. The queries to Spamhaus now come from the server's own IP, which is exactly what Spamhaus wants.

Check the config before you restart, so a typo does not take DNS down for the whole box:

```
unbound-checkconf
```

It must say no errors. Then:

```
systemctl restart unbound
```

Step 4: Verify — again, with the same two queries.

This step is where most tutorials stop and where mine starts.

```
dig +short 2.0.0.127.zen.spamhaus.org
```

```
dig +short 1.0.0.127.zen.spamhaus.org
```

First query: one or more 127.0.0.x answers. Second query: empty. Two different answers means the lookups are real again.

Then watch real mail come in. Send a test from an outside mailbox and follow the log:

```
tail -f /var/log/exim_mainlog
```

You want to see the message accepted, not rejected with in an RBL. In our case inbound mail from the big providers started flowing as soon as Unbound restarted.

Step 5: Leave the whitelist alone if it is there on purpose.

This server also had a WHM RBL whitelist for two Google mail ranges, 209.85.0.0/16 and 74.125.0.0/16. It was tempting to clean it up during the fix. We kept it. Another list, SpamCop, does sometimes list a real Gmail relay for a short time, and that whitelist keeps business mail flowing when it happens. It had nothing to do with this outage, so it stayed.

Plain takeaway: fix the cause you proved, and don't tidy up things you have not proven wrong in the same change.

The Bigger Lesson — A Mail Server Must Do Its Own Recursion

A web server can live happily behind a public resolver. A mail server cannot. Its spam defence depends on DNS questions that the blacklist operators only answer for resolvers they can trust to be one server, not millions of users.

Here is the short checklist I now run on any mail server I inherit:

- 1 **Test the RBL path with the test entries.** `dig +short 2.0.0.127.zen.spamhaus.org` must answer, `dig +short 1.0.0.127.zen.spamhaus.org` must not.
- 2 **Check the first hop.** `cat /etc/resolv.conf` should point to a local resolver.
- 3 **Check the second hop.** `grep -rn "forward" /etc/unbound/` should find no name: "." forwarder to public DNS.
- 4 **Validate before restart.** `unbound-checkconf` must say no errors.
- 5 **Search the logs for the symptom.** `grep -c "open resolver" /var/log/exim_mainlog` should be zero.
- 6 **Know why every whitelist entry exists.** Look at WHM › Exim Configuration Manager and write down the reason next to each RBL whitelist range.

Each item takes seconds. Together they turn "the whole internet is blacklisted" into a one-line fix.

What I Tell Every Customer Now

When a customer shows me their mail server design, I ask one question:

"When this server checks a blacklist, whose DNS resolver asks the question?"

If the answer is "the server's own resolver, doing its own recursion", they are in good shape. If the answer is "it uses Google DNS, it's faster", 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

What does "is in an RBL: Error: open resolver" mean?

It means your mail server asked a blacklist like Spamhaus through a public DNS resolver, and Spamhaus refused to answer. The refusal code starts with 127, so the mail server reads it as "listed" and rejects the sender. The sender is usually not listed at all.

Why does Spamhaus block 1.1.1.1 and 8.8.8.8?

Public resolvers carry queries for millions of users, so Spamhaus cannot tell who is asking or how much. It refuses those queries and returns 127.255.255.254 instead of a real answer. Mail servers should query from their own resolver and their own IP.

How do I test if my RBL lookups work?

Run `dig +short 2.0.0.127.zen.spamhaus.org` and `dig +short 1.0.0.127.zen.spamhaus.org` on the mail server. The first must return 127.0.0.x and the second must return nothing. If both give the same answer, your lookups are broken.

My resolv.conf already says 127.0.0.1. Can I still have this problem?

Yes. 127.0.0.1 only means a local resolver such as Unbound answers first. If that resolver has a forward-zone for "." pointing at public DNS, every query still leaves through the public resolver. Check the local resolver's own config.

Is it safe to remove the forward-zone from Unbound?

Yes, as long as the server can reach the internet on port 53. Unbound then resolves from the root servers itself. Back up the file, run `unbound-checkconf`, and restart the service, then repeat the two test queries.

Should I whitelist Google and Microsoft ranges to stop the rejections?

No, not for this problem. A whitelist only hides the symptom for the ranges you add, and every other sender stays blocked. Fix the resolver. Keep a whitelist only where you have a proven reason for it.

Read online: <https://quabin.com/spamhaus-open-resolver-all-mail-rejected/>