FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

DNS Health Says DKIM Is Green, but Nothing Is Signed — DKIM Signing Is a Separate Switch

A field note about a self-hosted mail server where every DNS check was green and every outgoing message was unsigned — the difference between a published DKIM record and a signed message — and the one test I now run before I call email authentication done.

Daanial Farrukh · 17 November 2026 · https://quabin.com/dkim-record-green-mail-not-signed-rspamd/

The admin panel looked perfect.

SPF: green. DKIM: green. DMARC: green.

The server was a self-hosted grommunio box on Ubuntu 24.04. Postfix in front, rspamd as the filter, a fresh domain with its records just published. The DKIM key had been generated in the admin interface, the TXT record added in DNS as dkim._domainkey.example.com, and the DNS Health page had turned it green within minutes.

"Email authentication is done," was the natural conclusion.

I asked one question instead: "Has anyone looked at the headers of a message this server actually sent?"

Nobody had.

So I sent one test message to a Gmail mailbox and one to a corporate mailbox behind a commercial spam filter, and opened the headers of both.

```
Authentication-Results: ... dkim=none header.d= ...
```

dkim=none. Not fail. Not pass. None. The receivers had found no DKIM signature on the message at all.

And the corporate filter had scored the message. One rule stood out: DMARC_FAIL_QUAR=6. The total was 6.524. The quarantine threshold on that filter was 6.6.

The message missed the quarantine folder by 0.076 points.

That's the trap. **A green DKIM check in DNS proves the record is published. It does not prove a single message was signed.**

What DKIM Actually Needs — Two Halves

DKIM has two halves, and they live in different places.

The public half is a TXT record in DNS. The private half is a key on the mail server that something must use to sign every outgoing message.

The flow, when it works:

Mail client → Postfix → milter (rspamd) → reads private key → adds DKIM-Signature header → internet → receiver looks up dkim._domainkey.example.com → verifies → dkim=pass

A DNS health check only tests the end of that chain: is the record there, is it valid, does it match the local key. It never sends a message. It cannot see whether the middle of the chain — the signer — is doing anything.

So you can have:

- A perfect public record and no signer configured at all.
- A signer configured but looking for the key in the wrong place.
- A signer looking in the right place but not allowed to read the file.
- A signer that works, but only for mail that passes through it — and some mail never does.

On this server, three of those four were true at the same time.

Where this setup is common:

- Self-hosted mail stacks where the admin UI generates the DKIM key but a separate filter (rspamd, OpenDKIM) does the signing.
- Servers built from a "publish these records" checklist, where the checklist ends at DNS.
- Any server where webmail submits mail locally instead of over SMTP.

Why the Usual Check Misses It

Three reasons, and they stack.

First, the dashboard is honest but narrow. It checks what it can check — DNS — and shows green. Nobody reads "DKIM: green" as "the record exists"; everyone reads it as "DKIM works".

Second, unsigned mail is usually still delivered. With SPF passing and a policy that is not yet at reject, most messages reach the inbox. Nothing bounces. Nobody complains. The signal is a score, not an error.

Third, published-but-unsigned is worse than no DKIM at all. A DMARC record tells the world "judge my mail by SPF and DKIM". With no signature, every message leans on SPF alone, and the first forward or mailing list breaks SPF alignment too. Then DMARC fails outright, and receivers do what the policy asks.

The only proof of signing is a received message. Everything else is a statement about DNS.

How to Actually Fix This

Everything up to Step 4 is read-only.

Step 1: Read the headers of a real received message.

Send a test message from the server to a mailbox you control at a large provider. In Gmail: open the message › three dots › Show original. Look at the Authentication-Results line.

```
dkim=none header.d=
```

That is the failure: no signature was found. What success looks like is dkim=pass with your domain in header.d= and your selector in header.s=.

Send one from webmail and one from a desktop client over SMTP. On this server, the difference between those two paths turned out to matter.

Step 2: Check whether rspamd has any signing config at all.

```
ls /etc/rspamd/local.d/
```

On this server the folder held only antivirus files and a controller file. There was no dkim_signing.conf. rspamd's DKIM module was loaded with its defaults, and the default key path is not where grommunio writes the key. grommunio's admin API writes each domain's key to:

```
/var/lib/grommunio-admin-api/<domain>.dkim.key
```

rspamd never looked there.

Step 3: Check whether rspamd can even read the key.

```
ls -ld /var/lib/grommunio-admin-api
```

The directory was 0750, owned by grommunio:gromox. The rspamd service user, _rspamd, is in neither group. Even with the right path, rspamd could not enter the directory, so it could not open the key.

Step 4: Check which mail actually passes through rspamd.

```
postconf smtpd_milters non_smtpd_milters
```

smtpd_milters was set to rspamd. non_smtpd_milters was empty.

Here's the part that trips people up. smtpd_milters only covers mail that arrives over SMTP. Webmail on this server submits through Postfix's local pickup service — the Received: header shows it coming from a local user id, not from a network connection. Mail that enters through pickup is "non-SMTP" mail, and with non_smtpd_milters empty it never reached rspamd. No scan, no signature.

Three faults. All three had to be fixed before a single message would be signed.

Step 5 (changes state): Point rspamd at the key.

Create /etc/rspamd/local.d/dkim_signing.conf:

```
path = "/var/lib/grommunio-admin-api/$domain.dkim.key";
selector = "dkim";
allow_username_mismatch = true;
use_domain = "header";
sign_authenticated = true;
sign_local = true;
```

$domain is filled in by rspamd per message, so one file covers every domain on the server. The selector must match the name of the DNS record — dkim here, because the record is dkim._domainkey.

Step 6 (changes state): Give rspamd read access with an ACL, not a group.

```
apt install -y acl
```

```
setfacl -m u:_rspamd:rx /var/lib/grommunio-admin-api

setfacl -d -m u:_rspamd:r /var/lib/grommunio-admin-api

setfacl -m u:_rspamd:r /var/lib/grommunio-admin-api/*.dkim.key
```

The second command is the important one. -d sets a **default** ACL on the directory, so every key file created there later — the next domain someone adds in the admin panel — is readable by rspamd from the moment it is written. Adding _rspamd to a group or running a one-off chmod fixes today's domain and silently breaks the next one.

Step 7 (changes state): Send locally submitted mail through rspamd too.

```
postconf -e 'non_smtpd_milters = inet:localhost:11332'
```

Use the same value your smtpd_milters already has. Then restart both services:

```
systemctl restart rspamd postfix
```

Step 8: Verify — with a received message, not a dashboard.

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

Send the same two test messages again: one from webmail, one over SMTP. Open the headers.

What success looked like on this server, at Gmail:

```
dkim=pass header.s=dkim
```

And at the corporate filter:

```
dkim=pass (2048-bit key) header.d=example.com
```

with DKIM_VALID_AU and DKIM_VALID_EF in the rule list. The spam score went from 6.524 to −2.096. Same server, same content, same IP. The only change was a signature.

What a failure looks like: still dkim=none means the message is not reaching rspamd or rspamd has no config for that domain. dkim=fail means a signature was added but does not verify — usually a selector that does not match the DNS record, or a record that was pasted with quotes, brackets or line breaks still in it. The record value must be one unbroken string.

Two More Faults the DKIM Work Exposed

Once you start reading what rspamd actually loaded instead of what the config files say, other things fall out. Two did here.

The antivirus file had been written with its own antivirus { } wrapper. Files in local.d are merged **into** the module section, so the wrapper created a rule inside a rule. rspamd logged cannot add rule and moved on. ClamAV was installed and running — and had never scanned a single message. The check that shows what rspamd really loaded:

```
rspamadm configdump antivirus
```

And clamd.conf had no TCPSocket line while rspamd was pointed at 127.0.0.1:3310. Installed, running and integrated are three different claims.

The second: outbound mail over IPv6 failed SPF. The SPF record listed only the IPv4 address, and the IPv6 address carried the provider's generic PTR. Until a correct reverse record and an ip6: term exist, the simple fix is to send over IPv4 only:

```
postconf -e 'inet_protocols = ipv4'
```

Neither of these would have shown up on a DNS health page either.

The Checklist I Run Now

Before I call email authentication done on any server:

1. Does a received message show dkim=pass with my domain? Gmail › Show original › Authentication-Results
2. Is there a signing config, and does its path match where the key really is? ls /etc/rspamd/local.d/
3. Can the signer read the key? ls -ld on the key directory, then getfacl on it
4. Does locally submitted mail pass the filter? postconf smtpd_milters non_smtpd_milters
5. Does a webmail message pass, not just an SMTP one? Test both paths
6. Is the antivirus rule actually loaded? rspamadm configdump antivirus
7. Does mail over IPv6 pass SPF? If not, inet_protocols = ipv4 until it does

Plain takeaway: **a green record is a promise. A signed message is the proof.**

What I Tell Every Customer Now

Whenever someone tells me their mail authentication is done, I ask one question:

"Can you show me the headers of a message your server sent today?"

If the answer is a header with dkim=pass and their own domain in it, they're in good shape. If the answer is "the dashboard is all green", we're not done yet — because the dashboard never sent a message.

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 does my DKIM check show green but emails show dkim=none?

A DNS check only verifies that the public DKIM record is published and valid. It never sends a message. dkim=none means the receiver found no signature at all, so the signer on your server is either not configured, cannot read the private key, or never sees the message.

What is the difference between dkim=none and dkim=fail?

dkim=none means the message carried no DKIM signature. dkim=fail means a signature was present but did not verify, usually because the selector does not match the DNS record or the record value was pasted with quotes, brackets or line breaks.

Why are webmail messages unsigned when SMTP messages are signed?

Many webmail systems submit mail locally through Postfix's pickup service instead of over SMTP. Postfix only sends that mail to milters listed in non_smtpd_milters. If only smtpd_milters is set, locally submitted mail is never scanned or signed.

Where does grommunio store the DKIM private key?

The grommunio admin API writes each domain's key to /var/lib/grommunio-admin-api/<domain>.dkim.key. That is not rspamd's default path, so rspamd needs a dkim_signing.conf in /etc/rspamd/local.d/ that points there, plus read access to the directory.

Why use an ACL instead of adding rspamd to a group?

A default ACL set with setfacl -d applies to every new file created in the directory. Each new domain added later gets a new key file, and the default ACL makes it readable by rspamd at once. A group change or chmod on today's file does not cover the next one.

How do I prove DKIM signing works?

Send a message from the server to a mailbox at a large provider and read the Authentication-Results header. You want dkim=pass with your own domain in header.d. Test both webmail and SMTP submission, because they can take different paths through the server.

Read online: https://quabin.com/dkim-record-green-mail-not-signed-rspamd/