FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

DKIM Is Green and Nothing Is Signed — Why a Published Record Is Not a Signature

A field note about a self-hosted mail server whose DNS health page showed DKIM in green while every message left unsigned — the three separate switches behind one green tick — and the one test I now trust instead of the dashboard.

Daanial Farrukh · 16 November 2026 · https://quabin.com/dkim-green-mail-not-signed-rspamd/

The server was a self-hosted grommunio box. New domain, new keys, everything set up through the admin panel.

The DKIM part looked done. The panel generated a 2048-bit keypair, gave me the TXT record, I published it in DNS, and the DNS health page turned DKIM green.

"So DKIM is working?"

That was the question I asked myself. The panel said yes.

Then I sent a test message to a Gmail address and opened "Show original". The authentication line said:

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

Not fail. Not pass. none. There was no signature on the message at all.

I sent another one to a mailbox on a second, unrelated mail system with a strict filter. Same result: no DKIM signature. And that filter showed what an unsigned message really costs. DMARC failed, the filter added 6 points for it, and the message scored 6.524 against a quarantine threshold of 6.6.

One tenth of a point from the spam folder. On a brand-new domain, from a server that was "green".

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

What the Green Tick Actually Checks

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

Publishing is DNS. A TXT record at <selector>._domainkey.example.com holds the public key. Anyone can look it up.

Signing is the mail server. For every outgoing message, some process must read the private key, hash the headers and body, and add a DKIM-Signature: header. The receiver then fetches the public key from DNS and checks the signature.

The flow when it works:

Webmail or client → Postfix → signer (rspamd) reads private key → adds DKIM-Signature → internet → receiver looks up selector._domainkey → dkim=pass

The flow on this server:

Webmail → Postfix → (no signer in the path) → internet → receiver finds no DKIM-Signature → dkim=none

A DNS health page can only test the first half. It looks up the TXT record and compares it with the key on disk. If they match, it is green. It never sees an outgoing message, so it cannot know whether anything signs with that key.

Where this gap is common:

- Self-hosted stacks where the admin panel creates keys but a separate filter (rspamd, OpenDKIM, Amavis) does the signing.
- Servers moved or rebuilt, where the DNS record survived but the signing config did not.
- Any setup checked only with a DNS lookup tool, never with a received message.

Why Nobody Sees It in Testing

Because almost every DKIM check people run is a DNS check.

dig TXT dkim._domainkey.example.com answers. Online DKIM checkers answer "valid record". The admin panel is green. All three are correct, and none of them proves signing.

And mail still gets delivered. Gmail accepted the test message. Without DKIM, a message can still pass DMARC through SPF alignment, and on most receivers the cost is a few spam points, not a bounce. Nothing breaks loudly. The mail just carries less trust than it should — until it meets one strict filter, or one domain with a tight DMARC policy, and lands in quarantine.

Only a received message proves signing. Everything else proves publishing.

How to Actually Fix This

On this server, three separate faults stopped signing. Any one of them alone would have been enough to leave mail unsigned. All three had to be fixed.

Steps 1 to 3 are read-only.

Step 1: Prove the problem with a real message.

Send a message from the server to a Gmail address. In Gmail, open the message, then the three-dot menu, then "Show original". Look at the Authentication-Results header.

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

That means no signature was found. If you see dkim=fail, a signature exists but does not verify — that is a different problem (wrong key, wrong selector, a header changed in transit). If you see dkim=pass header.d=example.com, signing already works and you can stop here.

Step 2: Check that the signer has a config at all.

grommunio uses rspamd as its filter. rspamd signs mail through its dkim_signing module, configured in local.d:

```
ls -l /etc/rspamd/local.d/dkim_signing.conf
```

On this server the file did not exist. rspamd was running, filtering, scoring — and signing nothing, because nobody had told it where the keys were.

You can also ask rspamd what it actually loaded:

```
rspamadm configdump dkim_signing
```

Step 3: Check that every outgoing path goes through the signer.

```
postconf smtpd_milters non_smtpd_milters
```

Here's the part that trips people up. Postfix has two milter settings. smtpd_milters covers mail that arrives over SMTP — a mail client submitting on port 587, or another server. non_smtpd_milters covers mail submitted locally, through the sendmail command and the pickup service.

Webmail on this server submitted locally. You can see it in the headers of a sent message: the first Received: line says the mail came in via local pickup from a user ID, not from an SMTP client. So every webmail message skipped rspamd completely, because non_smtpd_milters was empty.

Setting only smtpd_milters is the classic half-fix. Mail from Outlook gets signed, mail from webmail does not, and the test you happen to run decides what you believe.

Step 4: Check that the signer can read the key.

grommunio stores DKIM keys in its own directory, not in rspamd's default path:

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

On this server the directory was 0750, owned by grommunio:gromox. rspamd runs as the _rspamd user, which is neither the owner nor in the group. It could not even enter the directory, let alone read the key. A quick test as that user:

```
sudo -u _rspamd ls /var/lib/grommunio-admin-api
```

"Permission denied" here means rspamd will fail to load the key, even with a perfect config.

Step 5 (changes state): Create the signing config.

Create /etc/rspamd/local.d/dkim_signing.conf and point it at grommunio's key path and selector:

```
path = "/var/lib/grommunio-admin-api/$domain.dkim.key";
selector = "dkim";
```

$domain is filled in by rspamd per message, so one line covers every domain on the server. The selector must match the name of the published record — here, dkim._domainkey.example.com. Check yours with dig TXT <selector>._domainkey.example.com before you set it.

Step 6 (changes state): Send local mail through the milter too.

```
postconf -e 'non_smtpd_milters = $smtpd_milters'
```

This makes locally submitted mail use the same milter list as SMTP mail, so webmail, cron mail and scripts all pass through rspamd.

Step 7 (changes state): Give rspamd read access with an ACL, not a group.

The tempting fix is to add _rspamd to the gromox group. That gives rspamd far more than it needs. An ACL gives it exactly one thing: read access to that directory.

```
apt install acl

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

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

The first setfacl lets _rspamd enter and list the directory. The second is a **default** ACL: every key file created there later — when you add the next domain — is readable by _rspamd automatically. Without it, signing works today and quietly stops for the next domain you add.

If key files already exist, give them read access too: setfacl -m u:_rspamd:r /var/lib/grommunio-admin-api/*.dkim.key.

Then restart both services:

```
systemctl restart rspamd postfix
```

Step 8: Verify.

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

Send a new message from webmail — not from a desktop client, because webmail is the path that was broken — to Gmail and to any second mail system you can read. In "Show original" you want:

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

On this server, that is exactly what came back. And on the strict filter, the same kind of message went from 6.524 — one tenth of a point under quarantine — to -2.096. Same server, same content, same IP. The only change was a valid signature.

What a failure looks like: still dkim=none means the message did not pass through rspamd (check Step 6 and the Received: path again). dkim=fail or dkim=permerror means a signature was added but the selector or key does not match the published record (check Step 5 against dig).

Two More Silent Faults Found on the Same Server

Once I was reading rspamd's config, two more problems showed up. Neither had any error a user would see.

The virus scanner had never scanned anything. local.d/antivirus.conf repeated the outer antivirus { } wrapper. Files in local.d are merged *into* the module's section, so the wrapper created a nested block rspamd could not use. rspamd logged cannot add rule and moved on. ClamAV was installed and running, and no message was ever checked. On top of that, clamd.conf had no TCPSocket line while

rspamd was pointed at 127.0.0.1:3310. Check what rspamd really loaded:

```
rspamadm configdump antivirus
```

Outbound IPv6 failed SPF. The SPF record listed only the IPv4 address, and the IPv6 address had the hosting provider's default reverse DNS. Any message that left over IPv6 failed SPF. Until a proper IPv6 PTR and an ip6: term in SPF exist, the simple fix is to keep Postfix on IPv4:

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

Same lesson both times. **A component that is installed and running is not the same as a component that is doing its job.**

The Checklist I Run Now

For any self-hosted mail server, before I call deliverability done:

1. Is a signature on a received message? Gmail "Show original" → dkim=pass header.d=<your domain>
2. Does webmail mail pass, not just client mail? Send one of each and compare.
3. Does the signer have a config? rspamadm configdump dkim_signing
4. Does every path go through the milter? postconf smtpd_milters non_smtpd_milters
5. Can the signer read the key? sudo -u _rspamd ls <key directory>
6. Is the scanner really scanning? rspamadm configdump antivirus
7. Does every outbound address pass SPF? Check inet_protocols and the ip6: terms.

Plain takeaway: **test the message, not the record.**

What I Tell Every Customer Now

When a customer says their own mail server is "fully set up — SPF, DKIM, DMARC all green", I ask one question:

"Can you show me a message you received from that server today, with dkim=pass in its headers?"

If they can open one in a minute, they're in good shape. If the answer is "the dashboard shows green", we're not done yet — because the dashboard only knows what is in DNS, and DNS never sends a single email.

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 record show valid but emails say dkim=none?

A valid record only proves the public key is published in DNS. dkim=none means the received message carried no DKIM-Signature header, so nothing on the sending server signed it. Check that your signer (rspamd, OpenDKIM) has a signing config, can read the private key, and sees every outgoing message.

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

dkim=none means there was no signature on the message at all. dkim=fail means a signature was present but did not verify, usually because the selector or key does not match the published record, or the message was changed in transit. They have different fixes.

Why is webmail not DKIM-signed when Outlook mail is?

Webmail often submits mail locally through Postfix's sendmail command and pickup service, not over SMTP. Those messages only pass through a milter listed in non_smtpd_milters. If only smtpd_milters is set, mail from SMTP clients is signed and local webmail mail is not.

Where does rspamd look for DKIM keys on grommunio?

Not in rspamd's default location. grommunio's admin API stores keys as /var/lib/grommunio-admin-api/<domain>.dkim.key with the selector dkim. Set that path and selector in /etc/rspamd/local.d/dkim_signing.conf and make sure the _rspamd user can read the directory.

Should I add the rspamd user to the gromox group to read the keys?

An ACL is the narrower fix. setfacl -m u:_rspamd:rx on the directory gives read access to that one place, and a default ACL with setfacl -d -m u:_rspamd:r covers key files for domains added later. Group membership would give rspamd access to everything else the group can read.

How do I prove DKIM signing works?

Send a message from the server to a Gmail address and open "Show original". The Authentication-Results header should show dkim=pass with header.d set to your domain. Test webmail and a desktop client separately, because they can take different paths through Postfix.

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