emailverify.io

Stop rewriting the email. Run this one-hour list-health instead.

Bounces are not random. They are a specific, measurable, and fixable problem. Most teams are losing 2–5% of their sends to invalid addresses, stale records, and spam trap hits, and they don't know it. The underlying cause is almost always one of five things: how the list was built, how it has been cleaned (or not cleaned), how it is being sent to, how it is being verified, and how it is being monitored. This guide walks through each of those five causes, the specific fix for each, and the one-hour list-health check that ties them together. Every claim here is backed by a specific mechanism, not a generic "best practice" list. Read it once, run the checklist once, and you will know exactly which of the five things is broken in your list.

DAVID BRAMSON, CTO · 10 MIN READ · 2 JULY 2026

INTRODUCTION

Why this is for

- Signals that your list is rotting
- Why most advice fails
- Who this guide is for

Why bounces happen

- The real reason your emails don't get opened
- How email providers decide what lands in the inbox
- Hard bounces vs soft bounces
- The five address types hiding in your list

The fix

- The catch-all problem
- What happens when you send to a dirty list
- How email verification works
- Your pre-send checklist
- The list health audit
- Bring the file. We will return a list you can send to.

FAQS

Why this is for

Signals that your list is rotting

- You've noticed your emails don't get opened.
- Your bounce rate is creeping up, and you don't know why.
- You're about to send a campaign you can't afford to lose.
- You're about to scale sends and want to fix the issue before it gets worse.
- The team keeps blaming deliverability, and you want to see a rotting dataset yourself.

Why most advice fails

- Generic "best practices" lists are long, vague, and not specific to your list.
- Tools that only clean the list, without fixing what built it, will rot again within two months.
- The cause of a problem is almost never the one you first assume.

Who this guide is for

- Marketers who own a list and are responsible for deliverability.
- Founders who are about to scale sends and want to fix the issue before it gets worse.
- Ops people who inherited a list they didn't build and want to see what it is actually made of.

Why bounces happen

The real reason your emails don't get opened

You blame the copy because the copy is the part you can edit. The reason most copy, however, is not the problem is that the inbox provider never showed the email to the reader in the first place. Open rates are a function of inbox placement, sender reputation, and list quality. If the email never reaches the recipient's inbox, no amount of copy will change the open rate.

Inbox placement is decided by three things: the sender's domain reputation, the recipient's mail server's spam filtering, and the engagement history of similar emails to similar addresses. If your bounce rate is 4%, your sender reputation is being quietly downgraded across every mailbox you send to, and the next campaign inherits that downgrade. The copy is innocent. The list is not.

Spam traps are the clearest example of this. They are addresses that look like real mailboxes but exist only to catch senders who do not clean their lists. They are not bad people. They are not customers. They are a tripwire. Hitting one once is a warning. Hitting one repeatedly means the sender is being treated as a spammer by every major inbox provider.

Stale addresses are the slow version of the same problem. People change jobs, abandon old accounts, and stop checking inboxes they used to own. Mailbox providers track how long it has been since a given address last opened anything. Addresses that have been dormant for years are treated as high-risk by default, because they look like either a trap or a recycled address that is about to become one.

The fix is not better copy. The fix is a list that has been built, cleaned, and monitored in a way that keeps the sender's reputation intact. The rest of this guide is about exactly that.

How email providers decide what lands in the inbox

Mailbox providers decide one thing: do you want to send them what they want to receive. They do not care about your campaign. They care about their user. Every decision they make is a prediction about whether the recipient will open, read, and not complain about your email. Three signals feed that prediction: sender reputation, list quality, and engagement history. Get any one of the three wrong and the inbox will quietly stop showing your mail to its users.

Authentication

A message arrives at Gmail's inbox, and Gmail asks a simple question: is this sender who they claim to be? Three DNS records answer that question. SPF lists the servers allowed to send mail for the domain. DKIM adds a cryptographic signature that proves the message was not altered in transit. DMARC tells the receiver what to do when SPF or DKIM fails. Without all three, the sender looks like a spoofed domain, and most inboxes will treat it as spam or reject it outright.

Authentication is not optional anymore. Yahoo and Gmail both made it a requirement for bulk senders in 2024, and the rest followed within months. Setting up SPF, DKIM, and DMARC takes an hour and a DNS record per domain. Skipping it means every campaign you send is a coin flip that the inbox will accept the mail at all.

Reputation

Mailbox providers track every sender on a rolling 30-day window. Bounce rate, complaint rate, and spam trap hits are the three numbers that matter most. Cross the threshold on any one of them and the next campaign inherits a downgraded reputation by default. Recovery takes weeks, not days, and the only way to keep the reputation from dropping in the first place is to keep the list clean enough that the numbers never cross the line.

Engagement

Opens, clicks, replies, and time spent reading are the signals that tell a mailbox provider whether its users want this sender's mail. Senders who get opens and replies from a list will keep landing in the inbox. Senders who get deleted-without-opening will slowly lose the inbox, even if the list is technically clean and the authentication is perfect.

Hard bounces vs soft bounces

A hard bounce means the address is permanently invalid. The mailbox does not exist, the domain does not exist, or the recipient has been deactivated for years. Hard bounces are the cleanest signal in the system: the address will never accept mail again, and every attempt to send to it adds a small mark against the sender's reputation.

A soft bounce means the mailbox is temporarily unreachable. The inbox is full, the server is down, or the message is too large at the moment. Soft bounces are noisier and less dangerous than hard bounces, but a list that keeps generating soft bounces for months is usually a list that has stale addresses mixed in with the active ones.

Treat hard bounces as permanent and remove them immediately. Treat soft bounces as a pattern, not a single event: one soft bounce is noise, ten soft bounces from the same domain over a month is a problem worth investigating.

The five address types hiding in your list

Bounces are not random. They are the visible output of five underlying causes, each with a different fix. Most teams treat all bounces as one problem and wonder why the cleanup never sticks. Separating the five causes is the first step in any list-health audit.

	Hard bounce	Soft bounce
Invalid	An address that never existed, or one that was typed wrong at signup. Typos, fake addresses, and bots fall here.	Rarely a soft bounce. The address is valid, so the mailbox responds with a hard bounce instead.
Disposable	Throwaway addresses from temporary mail services. Most will bounce within days as the service rotates domains.	Temporary disposable services occasionally return soft bounces when their mail server is overloaded.
Role-based	Shared inboxes like info@, sales@, and support@. Low engagement and high complaint rates make them a quiet problem even when they do not bounce.	Rarely a soft bounce. The address is valid, so the mailbox responds with a hard bounce instead.
Catch-all	Domains that accept mail for any address, including ones that do not exist. Bounces arrive later, if at all, and only when the underlying mailbox rejects the message.	Soft bounces from catch-all domains usually mean the underlying mailbox is full or temporarily unreachable.
Spam trap	Addresses operated by ISPs and anti-spam vendors to catch senders who do not clean their lists. Hitting one once is a warning. Hitting one repeatedly means the sender is being treated as a spammer.	A trap that accepts mail can still flag the sender's domain for poor list hygiene. Soft bounces from traps are rare but worth noticing.

These five behaviors are different enough that they need to be handled separately. Invalid addresses are the easiest to remove: they never accept mail, so every attempt is wasted volume. Disposable addresses are a moving target, because the domains rotate faster than any static blocklist can keep up. Role-based addresses are technically valid but usually low-value, because the inbox is shared and the engagement is predictably low. Catch-alls are the most expensive to deal with, because the bounce arrives late or never, and the only way to know is to send and wait. Spam traps are the most dangerous: one hit can damage the sender's reputation across every mailbox they send to, and the damage compounds if the list keeps feeding them.

EXAMPLE

Run a list through a verifier and you will see all five in one report. Invalid addresses bounce immediately. Disposable and role-based addresses get flagged as low-value or high-risk. Catch-alls are marked separately, because the verifier cannot tell whether they will accept the mail until it actually arrives. Spam traps are flagged when the verifier recognises them from its own network. A clean report means none of the five are present in the list. A dirty report tells you which one is the problem, so the fix can be specific instead of guessing.

The fix

The catch-all problem

Catch-alls are the most expensive bounces in any list. They look like valid addresses, because the mailbox provider accepts mail for them at the SMTP level, but the underlying mailbox does not exist or has been deactivated. The bounce arrives later, if at all, and the sender has already paid the cost of sending it.

Catch-all configurations are common in SMB and in enterprise mail servers. A large company often runs a catch-all on its main domain so that typos in employee addresses do not lose mail. The same setup means every invalid address on that domain looks valid to a verifier. Some providers can detect catch-alls by probing with a second, random address and checking whether both are accepted. Others can only flag the domain as catch-all and leave the rest to the sender. Knowing which kind of verifier you are using matters here: a verifier that cannot detect catch-alls will let most of them through, and the bounce rate will look worse than the list actually is.

Why most verifiers stop at the catch-all

Detecting a catch-all requires sending a probe to a random address and waiting to see whether the server accepts it. Many verifiers skip this step because it adds latency and complexity to every check. The result is a report that marks catch-alls as "valid" and passes them through to the campaign. The sender then sees a bounce rate that the verifier never predicted, and the blame lands on the list rather than on the tool.

What happens when you send to a dirty list

Bad lists do not fail one bounce at a time. They fail in a single moment. The day the list crosses a threshold, mailbox providers stop treating the sender as a low-risk source and start treating every campaign as a potential spam run. Reading the impact from the graph:

1% bounce	Inbox providers notice. The sender is still in the green, but the trend is visible to anyone who looks.
2% bounce	Warnings appear in the dashboard. Gmail and Yahoo start flagging the domain in their internal systems.
3% bounce	Spam folder placement begins for a meaningful share of the list. Open rates drop across every campaign.
4% bounce	Sending limits start to be applied. Campaigns that used to go out in full are throttled or split across days.
5% bounce	Blocklists and major inbox providers treat the domain as a spam source. Recovery takes weeks, not days.

The chart assumes a typical B2B list with a healthy opt-in history. A list with a high share of purchased or scraped addresses will hit the same thresholds faster, because the underlying bounce rate is higher to begin with. The lesson is that the threshold is not a fixed number. It is the point at which the sender's reputation can no longer absorb the next campaign without damage, and that point moves depending on how much clean mail the sender has already sent.

EXAMPLE

A marketing team runs a 40,000-contact list that has been untouched for three years. Their last campaign went out to all of it and the bounce rate came back at 6%. Gmail started routing their next campaign to spam within the same week. The list was not malicious, and the team did not buy anything. It was just old enough that a meaningful share of the addresses had been deactivated, recycled, or never real in the first place. The fix was not better copy. The fix was a verification pass and a re-permission campaign before any more mail went out.

How email verification works

Email verification is a sequence of checks, each one ruling out a different kind of bad address. A single check misses the majority of the problem. Stacking the checks in the right order catches most of what any one check would let through.

Stage 1. Syntax

The address must be parsable against the RFC 5322 and RFC 6531 specifications. Malformed characters, missing @ signs, and illegal Unicode in the local part all fail here. Syntax is the cheapest check and catches the obvious typos before any more expensive lookup runs.

Stage 2. DNS / MX lookup

The domain in the address must resolve to a real DNS record, and that record must point to a mail server. Domains that have been deleted, expired, or never registered fail here. Typos in the domain part often land here, because the typed domain does not exist anywhere on the internet.

Stage 3. SMTP handshake

The verifier opens a connection to the mail server and asks whether it would accept mail for the address, without actually sending a message. Most servers answer within a few seconds. A bounce here means the address is rejected at the server level, before any message is sent. Some servers accept everything regardless, which is why the next stage matters.

Stage 4. Catch-all detection

The verifier sends a second, random address to the same domain. If both addresses are accepted, the domain is catch-all and the original address cannot be confirmed from the server alone. Some verifiers skip this step and report the domain as "valid". Others flag it as "catch-all" so the sender can decide what to do with it.

Stage 5. Disposable and role-based checks

Disposable domains are matched against a rotating list of temporary mail services. Role-based addresses are matched against a short list of shared inboxes such as info@, sales@, and support@. Both are technically valid addresses and both are low-value for most senders.

Stage 6. Spam trap and blocklist checks

Spam trap hits are matched against known trap networks. Blocklist checks look up the sending domain and the recipient domain against public and private blocklists. Hits here are the most expensive signal in the system, because one hit can damage the sender's reputation across every mailbox they send to.

Your pre-send checklist

Run this checklist against any list before it goes out. Each item takes a few minutes to run and prevents a specific problem.

- **Remove hard bounces.** Invalid addresses, dead domains, and mailboxes that have been closed for years.
- **Remove spam traps.** Single-purpose trap addresses and known trap domains.
- **Remove disposable addresses.** Throwaway mail services that will bounce within days.
- **Flag catch-alls for a separate review.** Do not mix them into the main campaign.
- **Suppress role-based addresses.** info@, sales@, support@, and similar shared inboxes.
- **Re-verify any address older than six months.** Mailbox providers track how long it has been since a given address last opened anything.
- **Confirm SPF, DKIM, and DMARC are live on the sending domain.** Authentication is the prerequisite for inbox placement at all.
- **Start a small test send before the full campaign.** A few hundred recipients is enough to see the bounce rate before it matters.
- **Watch the bounce rate for the first 24 hours after the send.** If it crosses 2%, pause the campaign and investigate before continuing.

The list health audit

Run this once a quarter. It takes one hour and catches the slow rot that accumulates between campaigns. Each step has a specific output that tells you what to do next.

Step 1. Inventory

Export the list and count the total. Note the age of the oldest contact, the last time it was cleaned, and the last time it was sent to. Lists that have not been touched in two years are usually the ones that bounce the most. The inventory gives you the baseline to compare against after the cleanup.

Step 2. Segment by engagement

Split the list into three groups: opened in the last 90 days, opened in the last 365 days, and never opened. The first group is the safest to send to. The second group is the middle ground. The third group is the most expensive to keep and the most likely to bounce or complain when it does.

Step 3. Run a verification pass

Run the full list through a verifier that runs the six stages described in the section above. Export the results as a CSV with one row per address and one column per check.

Step 4. Remove the unsendables

Remove everything that failed syntax, DNS, or SMTP. Remove everything that hit a spam trap or a known blocklist. Remove everything that is disposable or role-based unless the campaign specifically needs it. Keep the catch-alls in a separate file for a manual review.

Step 5. Re-permission the engaged

Send one last email to the never-opened group and the two-year-old group asking them to confirm they want to receive mail from the sender. Anyone who does not click within a week is removed. This is the step that turns a dirty list into a list you can actually send to without damaging the sender's reputation.

Step 6. Schedule the next run

Set a calendar reminder for the same checklist four weeks from now. Lists rot between campaigns, and the only way to keep the bounce rate from drifting back up is to run the checklist on a regular cadence.

Bring the file. We will return a list you can send to.

If you'd rather not run the checklist yourself, bring the file and we will return a list you can send to. Upload your list, we run the six-stage verification, and you get back a CSV with one row per address and one column per check. Hard bounces, spam traps, disposables, and catch-alls are flagged so you can decide what to do with each one.

FREE TIER · 100 CREDITS · NO CARD · NO EXPIRY

Start a free trial at app.emailverify.io, with 100 credits and no card required. Pay-as-you-go credits start at $1.99 per 1,000 verifications, and the credits do not expire while you are testing the workflow.

VERIFYIO

Verifyio is a B2B email verification platform for teams that need to know which addresses will bounce before they send. Six-stage verification, catch-all detection, disposable and role-based flags, spam trap and blocklist checks, and a CSV export that drops into any campaign tool. Pay-as-you-go credits, no subscription required to start.

START FREE TRIAL →

FAQS

How often should I run a list through verification?

Once a quarter for a stable list, and before every large campaign if the list has been untouched for more than two months. Lists that are mailed weekly need more frequent checks than lists that are mailed once a year. The cadence should match how fast the list is being used, not how fast the calendar moves.

Do I need to remove every catch-all address?

Not automatically. Catch-alls are the most expensive bounces in any list, because the bounce arrives late or never, and the only way to know is to send and wait. Most teams keep a small share of catch-alls in the list and monitor their bounce rate closely for the first few sends to those addresses.

What is the difference between a hard bounce and a soft bounce?

A hard bounce means the address will never accept mail again: the mailbox does not exist, the domain does not exist, or the recipient has been deactivated for years. A soft bounce means the mailbox is temporarily unreachable: the inbox is full, the server is down, or the message is too large at the moment. Hard bounces should be removed immediately. Soft bounces are noisier and less dangerous, but a list that keeps generating soft bounces for months is usually a list that has stale addresses mixed in with the active ones.

How long does a verification pass take?

Most verifiers return results within a few seconds per address for the syntax, DNS, and SMTP checks. Catch-all detection adds a second probe per domain and slows the run down by roughly a factor of two. Spam trap and blocklist checks are usually instant, because they are matched against a cached list rather than a live lookup.

Can I run this on a list I bought?

Yes, and you should run it before any mail goes out. Purchased lists are the most expensive lists to send to, because the bounce rate is usually high and the spam trap hits are common. Bring the file, run the six-stage verification, remove everything that failed, and start with a small test send before the full campaign.

Does this work for transactional email?

Yes, the same checks apply to password resets, receipts, and any other mail that relies on the recipient's inbox. Transactional mail is usually more time-sensitive than marketing mail, so the bounce rate matters even more because a missed password reset is a support ticket.